

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DD
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KKKKKK  KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DD      DD
KKKKKK  KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DD      DD
KK      KK  EE          YY      YY  PP      PP  AAAAAAAAAA DD      DD
KK      KK  EE          YY      YY  PP      PP  AAAAAAAAAA DD      DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD
KK      KK  EEEEEEEEE  YY      YY  PP      PP  AA      AA  DDDDDDD  DD
KK      KK  EEEEEEEEE  YY      YY  PP      PP  AA      AA  DDDDDDD  DD
                                     ....
                                     ....
                                     ....
                                     ....
```

```
LL      IIIII  SSSSSSS
LL      IIIII  SSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSS
LL      II     SSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLL IIIII  SSSSSSS
LLLLLLLL IIIII  SSSSSSS
```

KEYPAD
Table of contents

M 3
- KEYPAD SYMBOL TABLE MANIPULATION ROUTI 15-SEP-1984 23:59:38 VAX/VMS Macro V04-00

Page 0

(3)	147	DEFINE KEYPAD SYMBOL
(4)	295	DELETE KEYPAD SYMBOL
(7)	465	SHOW KEYPAD SYMBOL TABLE ENTRIES
(10)	815	ALLOCATE AND INSERT ENTRY IN KEYPAD SYMBOL TABLE
(11)	920	CHECK FOR SYNONYM KEY NAMES
(12)	998	SEARCH FOR SYMBOL ENTRY IN KEYPAD SYMBOL TABLE
(13)	1035	SEARCH KEYPAD SYMBOL TABLE FOR ENTRY
(14)	1110	SET KEYPAD STATE
(15)	1184	ALLOCATE AND INIT A KEYPAD STATE SYMBOL
(16)	1223	DEALLOCATE A KEYPAD STATE SYMBOL

```
0000 1 .TITLE KEYPAD - KEYPAD SYMBOL TABLE MANIPULATION ROUTINES
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 * ALL RIGHTS RESERVED.
0000 9
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 * TRANSFERRED.
0000 16
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 * CORPORATION.
0000 20
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23
0000 24 *****
0000 25
0000 26 KEYPAD SYMBOL TABLE MANIPULATION ROUTINES
0000 27
0000 28
0000 29 AUTHOR: Peter George 15-March-1983
0000 30
0000 31 These routines all assume the structure of a keypad symbol
0000 32 block, starting with the symbol name (SYM_T_SYMBOL), is as follows:
0000 33
0000 34 Byte - length of symbol name
0000 35 ASCII symbol name
0000 36 Word - combined lengths of next three strings + 4
0000 37 Byte - length of if_state string
0000 38 ASCII if_state string
0000 39 Word - length of symbol value
0000 40 ASCII symbol value
0000 41 Byte - length of set_state string
0000 42 ASCII set_state string
0000 43
0000 44 MODIFIED BY:
0000 45
0000 46 V03-006 HWS0058 Harold Schultz 18-Apr-1984
0000 47 Use DCL$ALLOC STATE when setting temporary key states
0000 48 and DCL$LOCKED_STATE to restore original state.
0000 49 Exchange the synonym keypad names with the common key names.
0000 50 (i.e. translate "FIND" to "E1" instead of "E1" to "FIND")
0000 51
0000 52 V03-006 HWS0052 Harold Schultz 09-Apr-1984
0000 53 Translate synonym keypad names to a common key name.
0000 54 (i.e. Translate "E1" to "FIND")
0000 55 Add SHOW KEY/LOG.
0000 56
0000 57 V03-005 PCG0005 Peter George 09-Feb-1984
```


0000	58	:	Change format of SHOW KEY display.
0000	59	:	Add SHOW KEY/FULL.
0000	60	:	Zero the input buffer at the end of DCL\$DEFKEY.
0000	61	:	
0000	62	:	V03-004 PCG0004 Peter George 01-Dec-1983
0000	63	:	Add /ERASE.
0000	64	:	
0000	65	:	V03-003 PCG0003 Peter George 27-Jul-1983
0000	66	:	Move PSECT declaration.
0000	67	:	
0000	68	:	V03-002 PCG0002 Peter George 27-May-1983
0000	69	:	Validate key names before defining them.
0000	70	:	
0000	71	:	V03-001 PCG0001 Peter George 07-Apr-1983
0000	72	:	Tolerate omission of SET KEY qualifiers.
0000	73	:	Add SHOW KEY/BRIEF/DIRECTORY.
0000	74	:---	

```
0000 76 :  
0000 77 : MACRO LIBRARY CALLS  
0000 78 :  
0000 79 : PRCDEF ;DEFINE PROCESS DATA STRUCTURE  
0000 80 : WRKDEF ;DEFINE COMMAND DATA STRUCTURE  
0000 81 : PTRDEF ;DEFINE TOKEN DESCRIPTORS  
0000 82 : SYMDEF ;DEFINE SYMBOL ENTRY OFFSETS  
0000 83 : $CLMSGDEF ;DEFINE ERROR/STATUS VALUES  
0000 84 : $STSDEF ;DEFINE STATUS LONGWORD  
0000 85 :  
0000 86 : .PSECT DCL$ZCODE,BYTE,RD,NOWRT  
0000 87 :  
0000 88 :  
0000 89 : ASCII TEXT STRINGS FOR SHOW KEYS DISPLAY.  
0000 90 :  
0000 91 : SHOWHDR:  
20 64 61 70 79 65 6B 20 43 41 21 00' 0000 92 : .ASCII '!AC keypad definitions:'  
3A 73 6E 6F 69 74 69 6E 69 66 65 64 000C  
17 0000  
0018  
41 21 22 20 3D 20 53 41 21 20 20 00' 0018 93 BRIEFFAO:  
22 53 0024 94 : .ASCII ' !AS = ''!AS''  
0D 0018  
0026  
41 21 22 20 3D 20 53 41 21 20 20 00' 0026 95 FULLFAO:  
6F 68 63 65 43 41 21 28 20 20 22 53 0032 96 : .ASCII ' !AS = ''!AS'' (!ACecho,!ACterminate,!ACerase,!AClock!AC!AC!AS)'  
74 61 6E 69 6D 72 65 74 43 41 21 2C 003E  
21 2C 65 73 61 72 65 43 41 21 2C 65 004A  
43 41 21 43 41 21 68 63 6F 6C 43 41 0056  
29 53 41 21 0062  
3F 0026  
6F 6E 00' 0066 97 NO: .ASCII 'no'  
02 0066  
3D 65 74 61 74 73 00' 0069 98 STATE: .ASCII 'state='  
06 0069  
00 0070 99 NULL: .BYTE 0  
2C 00' 0071 100 COMMA: .ASCII ', '  
01 0071  
0073  
0073 101 :  
0073 102 : SYNONYM KEY NAME TABLES  
0073 103 :  
0073 104 :  
0073 105 : DEFINE SYNONYM KEY NAMES  
0073 106 :  
0073 107 : SYNONYM NAME SETS UP THE RELATIONSHIP BETWEEN THE SYNONYM (NAME1) AND  
0073 108 : THE COMMON KEY NAME (NAME2) THAT THE SYNONYM IS TRANSLATED TO. IF A NEW  
0073 109 : SYNONYM IS CREATED THAT TRANSLATES TO AN EXISTING COMMON KEY NAME (IN  
0073 110 : SYNDEF_TAB), ONLY AN ENTRY IN SYNNAME_TAB NEEDS TO BE ADDED. IF A NEW  
0073 111 : COMMON KEY NAME IS NEEDED, THEN ADD IT TO SYNDEF_TAB.  
0073 112 :  
0073 113 : .MACRO SYNONYM_NAME NAME1,NAME2  
0073 114 : .ASCII 'NAME1'  
0073 115 : .WORD 'NAME2'_ADR - SYNDEF_TAB  
0073 116 : .ENDM  
0073 117 :  
0073 118 :  
0073 119 : SYNNAME_TAB:
```

```
0073 120      SYNONYM_NAME  FIND,E1      ;DEFINE SYNONYM KEY NAMES
007A 121      SYNONYM_NAME  INSERT_HERE,E2
0088 122      SYNONYM_NAME  REMOVE,E3
0091 123      SYNONYM_NAME  SELECT,E4
009A 124      SYNONYM_NAME  PREV_SCREEN,E5
00A8 125      SYNONYM_NAME  NEXT_SCREEN,E6
00B6 126      .BYTE        0      ;END OF TABLE MARKER
00B7 127      :
00B7 128      : DEFINE COMMON SYNONYM KEY NAMES
00B7 129      :
00B7 130      : NAME = COMMON TRANSLATED NAME (I.E. "FIND")
00B7 131      :
00B7 132      : .MACRO  SYNONYM_TRN      NAME
00B7 133      'NAME' _ADR:
00B7 134      .ASCIC  "NAME"
00B7 135      .ENDM
00B7 136
00B7 137
00B7 138      SYNDEF_TAB:
00B7 139      SYNONYM_TRN      E1
00BA 140      SYNONYM_TRN      E2
00BD 141      SYNONYM_TRN      E3
00C0 142      SYNONYM_TRN      E4
00C3 143      SYNONYM_TRN      E5
00C6 144      SYNONYM_TRN      E6
00C9 145
```



```
00C9 147 .SBTTL DEFINE KEYPAD SYMBOL
00C9 148 :+
00C9 149 : DCL$DEFKEY - DEFINE KEYPAD SYMBOL
00C9 150 :
00C9 151 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DEFINE/KEY
00C9 152 : COMMAND.
00C9 153 :
00C9 154 : INPUTS:
00C9 155 :
00C9 156 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
00C9 157 : R9 = ADDRESS OF SCRATCH STACK.
00C9 158 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
00C9 159 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
00C9 160 :
00C9 161 : OUTPUTS:
00C9 162 :
00C9 163 : THE SPECIFIED META-KEY IS ASSIGNED TO THE SPECIFIED EQUIVALENCE
00C9 164 : STRING.
00C9 165 :-
00C9 166 :
00C9 167 DCL$DEFKEY:: ;DEFINE META-KEY EQUIVALENCE
00C9 168 :
00C9 169 :
00C9 170 : SET INITIAL PARSE STATE.
00C9 171 :
56 01 D0 00C9 172 : MOVL #SYM_M_ECHO,R6 ;INITIALIZE KEYPAD FLAGS
57 01 D0 00C9 173 : MOVL #1,R7 ;INITIALIZE LOCAL FLAGS (/LOG)
58 D4 00CF 174 : CLRL R8 ;CLEAR IF_STATE TOKEN PTR
79 7C 00D1 175 : CLRQ -(R9) ;ALLOCATE SET_STATE DESCRIPTOR
00D3 176 :
00D3 177 :
00D3 178 : PROCESS THE TOKENS ON THE COMMAND LINE.
00D3 179 :
FF2A' 30 00D3 180 : BSBW DCL$GETDVAL ;SKIP PAST /KEY DESCRIPTOR
FF27' 30 00D6 181 10$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUES
55 04 91 00D9 182 : CMPB #PTR_K_ENDLINE,R5 ;EOL?
03 12 00DC 183 : BNEQ 15$ ;NO, CONTINUE PARSING
00AF 31 00DE 184 : BRW 70$ ;YES, THEN DONE PARSING
55 03 91 00E1 185 15$: CMPB #PTR_K_PARAMETR,R5 ;ITEM TYPE PARAMETER?
05 12 00E4 186 : BNEQ 20$ ;NO, THEN PROCESS QUALIFIER
79 51 7D 00E6 187 : MOVQ R1, -(R9) ;SAVE PARAMETER DESCRIPTOR
EB 11 00E9 188 : BRB 10$ ;GET NEXT TOKEN
FF12' 30 00EB 189 20$: BSBW DCL$GETNVAL ;GET QUALIFIER NUMBER
00000000'8F 51 D1 00EE 190 : CMPL R1, #CLISK_DEFK_TERM ;QUALIFIER MATCH?
38 13 00F5 191 : BEQL 30$ ;YES, THEN PROCESS
00000000'8F 51 D1 00F7 192 : CMPL R1, #CLISK_DEFK_ECHO ;QUALIFIER MATCH?
3B 13 00FE 193 : BEQL 35$ ;YES, THEN PROCESS
00000000'8F 51 D1 0100 194 : CMPL R1, #CLISK_DEFK_LOCK ;QUALIFIER MATCH?
3E 13 0107 195 : BEQL 40$ ;YES, THEN PROCESS
00000000'8F 51 D1 0109 196 : CMPL R1, #CLISK_DEFK_LOG ;QUALIFIER MATCH?
41 13 0110 197 : BEQL 43$ ;YES, THEN PROCESS
00000000'8F 51 D1 0112 198 : CMPL R1, #CLISK_DEFK_SET_ ;QUALIFIER MATCH?
4F 13 0119 199 : BEQL 45$ ;YES, THEN PROCESS
00000000'8F 51 D1 011B 200 : CMPL R1, #CLISK_DEFK_IF_S ;QUALIFIER MATCH?
58 13 0122 201 : BEQL 50$ ;YES, THEN PROCESS
00000000'8F 51 D1 0124 202 : CMPL R1, #CLISK_DEFK_ERAS ;QUALIFIER MATCH?
31 13 012B 203 : BEQL 55$ ;YES, THEN PROCESS
```



```

      A7 11 012D 204 25$: BRB 10$ :GET NEXT
      012F 205
      012F 206 30$: SETBIT SYM V TERMINATE,R6 :ASSUME /TERMINATE
F7 53 00 E1 0132 207 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /NOTERMINATE
      0136 208 CLRBIT SYM V TERMINATE,R6 :CLEAR TERMINATE FLAG
      F2 11 0139 209 BRB 25$ :GET NEXT
      013B 210 35$: SETBIT SYM V ECHO,R6 :ASSUME /ECHO
EB 53 00 E1 013E 211 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /NOECHO
      0142 212 CLRBIT SYM V ECHO,R6 :CLEAR ECHO FLAG
      E6 11 0145 213 BRB 25$ :GET NEXT
      0147 214 40$: SETBIT SYM V LOCK,R6 :ASSUME /LOCK
DF 53 00 E1 014A 215 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /NOLOCK
      014E 216 CLRBIT SYM V LOCK,R6 :CLEAR LOCK FLAG
      DA 11 0151 217 BRB 25$ :GET NEXT
      57 01 D0 0153 218 43$: MOVL #1,R7 :ASSUME /LOG
D3 53 00 E1 0156 219 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /NOLOG
      57 D4 015A 220 CLRL R7 :CLEAR FLAG
      CF 11 015C 221 BRB 25$ :GET NEXT
      015E 222 55$: SETBIT SYM V ERASE,R6 :ASSUME /ERASE
CB 53 00 E1 0161 223 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /NOERASE
      0165 224 CLRBIT SYM V ERASE,R6 :CLEAR ERASE FLAG
      C3 11 0168 225 BRB 25$ :GET NEXT
      016A 226
      016A 227 45$: CLRBIT SYM V STATE,R6 :ASSUME /NOSET_STATE
BC 53 00 E0 016D 228 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /SET_STATE
      0171 229 SETBIT SYM V STATE,R6 :SET STATE FLAG
      FE89' 30 0174 230 BSBW DCL$GETDVAL :GET THE ASSOCIATED VALUE
      69 51 7D 0177 231 MOVQ R1,(R9) :SAVE THAT VALUE
      B1 11 017A 232 BRB 25$ :GET NEXT
      017C 233
      58 D4 017C 234 50$: CLRL R8 :ASSUME /NOIF STATE
AB 53 00 E0 017E 235 BBC #PTR V NEGATE-PTR_V_FLAGS,R3,25$ :IGNORE IF NOT /IF STATE
      BA AA D0 0182 236 MOVL WRK C RSLNXT(R10),R8 :SAVE VALUE TOKEN PTR
      FE77' 30 0186 237 52$: BSBW DCL$GETDVAL :GET NEXT DESCRIPTOR VALUE
      54 05 D1 0189 238 CMPL #PTR_K_COMMA,R4 :TERMINATOR A COMMA?
      F8 13 018C 239 BEQL 52$ :GET NEXT VALUE
      9D 11 018E 240 BRB 25$ :GET NEXT
      0190 241
      0190 242 :
      0190 243 : INSERT THE META-KEY SYMBOL IN THE SPECIFIED KEYPAD SYMBOL TABLES.
      0190 244 :
      0190 245 : SCRATCH STACK LOOKS LIKE:
      0190 246 : (R9) EQUIVALENCE STRING DESCRIPTOR
      0190 247 : 8(R9) META-KEY NAME DESCRIPTOR
      0190 248 : 16(R9) SET STATE DESCRIPTOR
      0190 249 : R6 CONTAINS SYMBOL FLAGS
      0190 250 :
      08 A9 7F 0190 251 70$: PUSHQ 8(R9) :PUSH THE DESCRIPTOR ADDRESS
00000000'EF 01 FB 0193 252 CALLS #1,VALIDATE_KEY_NAME :IS IT VALID?
      5F 50 E9 019A 253 BLBC R0,97$ :NO, THEN RETURN ERROR
      50 57 D0 019D 254 MOVL R7,R0 :GET /LOG FLAG
      51 08 A9 7D 01A0 255 MOVQ 8(R9),R1 :GET KEY NAME DESCRIPTOR
      04AA 30 01A4 256 BSBW DCL$SYNONYM :CHECK FOR SYNONYMS
      08 A9 51 7D 01A7 257 MOVQ R1,8(R9) :SAVE RETURNED KEYPAD NAME
      03 56 01 E0 01AB 258 BBC #SYM V TERMINATE,R6,71$ :BRANCH IF /TERMINATE
      56 01 CA 01AF 259 BICL #SYM M ECHO,R6 :IGNORE THE ECHO FLAG
      01B2 260 71$: ASSUME PTR_K_COMMA NE 0
```

```
BA AA 58 DO 01B2 261 MOVL R8,WRK_L_RSLNXT(R10) ;RESET FOR FIRST IF_STATE VALUE
      09 13 01B6 262 BEQL 75$ ;SKIP IF NONE
      FE45' 30 01B8 263 72$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUE
58 54 DO 01BB 264 MOVL R4,R8 ;SAVE THE TERMINATOR
      05D3 30 01BE 265 BSBW DCL$ALLOC_STATE ;SET NEW STATE
      03F9 30 01C1 266 75$: BSBW DCL$ALLOCKEY ;ALLOCATE THE KEYPAD SYMBOL
      31 50 E9 01C4 267 BLBC R0,95$ ;BRANCH IF ERROR
      01C7 268
      01C7 269
      01C7 270 ; OUTPUT /LOG MESSAGE.
      01C7 271
      13 57 E9 01C7 272 BLBC R7,80$ ;SKIP IF /NOLOG
      08 A9 9F 01CA 273 PUSHAB 8(R9) ;SET ADDRESS OF META-KEY NAME DESCR
      48 AB DD 01CD 274 PUSHL PRC_L_CURRKEY(R11) ;SET ADDRESS OF ASCII STATE NAME
50 51 02 DO 01D0 275 MOVL #2,R1 ;SET ARGUMENT COUNT
      0003DDC3 8F DO 01D3 276 MOVL #CLIS_DEFKEY,R0 ;SET STATUS
      FE23' 30 01DA 277 BSBW DCL$FORMMSG ;OUTPUT THE LOG MESSAGE
      58 05 D1 01DD 278 80$: CMPL #PTR_K_COMMA,R8 ;TERMINATOR A COMMA?
      05 12 01E0 279 BNEQ 90$ ;NO, TIME TO EXIT
      FE1B' 30 01E2 280 BSBW DCL$LOCKED_STATE ;YES, RESTORE LOCKED KEY STATE
      D1 11 01E5 281 BRB 72$ ; BEFORE GETTING NEXT STATE
      01E7 282
      01E7 283 ;
      01E7 284 ; RESTORE KEYPAD STATE AND RETURN.
      01E7 285
      01E7 286
0100 8F 00 6E 00 2C 01E7 287 90$: MOVCS #0,(SP),#0,#WRK_C_INPBUFSIZ,- ;RESET THE INPUT BUFFER
      F896 CA 01EE 288 WRK_G_INPBUF(R10) ;
      FE05' 30 01F8 289 STATUS NORMAL ;SET NORMAL COMPLETION
      05 01FB 290 95$: BSBW DCL$LOCKED_STATE ;RESTORE KEY STATE
50 00038280 8F DO 01FC 291 RSB
      F3 11 0203 292 97$: MOVL #CLIS_IVKEYNAM,R0 ;SET STATUS
      293 BRB 95$ ;RETURN
```

```

0205 295 .SBTTL DELETE KEYPAD SYMBOL
0205 296
0205 297 *
0205 298 DCL$DELKEY - DELETE KEYPAD SYMBOL
0205 299 THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DELETE/KEY
0205 300 COMMAND.
0205 301
0205 302 INPUTS:
0205 303
0205 304 R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0205 305 R9 = ADDRESS OF SCRATCH STACK.
0205 306 R10 = BASE ADDRESS OF COMMAND WORK AREA.
0205 307 R11 = BASE ADDRESS OF PROCESS WORK AREA.
0205 308
0205 309 OUTPUTS:
0205 310
0205 311 THE SPECIFIED META-KEY IS DELETED FROM THE SYMBOL TABLE.
0205 312
0205 313
0205 314 DCL$DELKEY:: ;DELETE KEYPAD DEFINITION
0205 315
0205 316 SET INITIAL PARSE STATE.
0205 317
0205 318 CLRL R8 ;CLEAR STATE TOKEN PTR
0205 319 PUSHL #3 ;INITIALIZE LOCAL FLAGS
0205 320 ; /LOG, /ALL, NO UNDEFINED SYMBOLS
0205 321
0205 322
0205 323 PROCESS THE TOKENS ON THE COMMAND LINE.
0205 324
0205 325
0205 326 10$: BSBW DCL$GETDVAL ;SKIP PAST /KEY DESCRIPTOR
0205 327 BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUES
0205 328 CMPB #PTR_K_ENDLINE,R5 ;EOL?
0205 329 BNEQ 15$ ;NO, CONTINUE PARSING
0205 330 BRW 50$ ;YES, THEN DONE PARSING
0205 331 15$: CMPB #PTR_K_PARAMETR,R5 ;ITEM TYPE PARAMETER?
0205 332 BNEQ 20$ ;NO, THEN PROCESS QUALIFIER
0205 333 MOVQ R1,R6 ;SAVE PARAMETER DESCRIPTOR
0205 334 BICL #2,(SP) ;CLEAR /ALL FLAG
0205 335 BRB 10$ ;GET NEXT TOKEN
0205 336 20$: BSBW DCL$GETNVAL ;GET QUALIFIER NUMBER
0205 337 CMPL R1,#CLISK_DELK_LOG ;QUALIFIER MATCH?
0205 338 BEQL 25$ ;YES, THEN PROCESS
0205 339 CMPL R1,#CLISK_DELK_STAT ;QUALIFIER MATCH?
0205 340 BEQL 30$ ;YES, THEN PROCESS
0205 341 BRB 10$ ;GET NEXT
0205 342 25$: BISL #1,(SP) ;ASSUME /LOG
0205 343 BBC #PTR_V_NEGATE-PTR_V_FLAGS,R3,10$ ;IGNORE IF NOT /NOLOG
0205 344 BICL #1,(SP) ;CLEAR FLAG
0205 345 BRB 10$ ;GET NEXT
0205 346
0205 347 30$: CLRL R8 ;ASSUME /NOSTATE
0205 348 BBS #PTR_V_NEGATE-PTR_V_FLAGS,R3,10$ ;IGNORE IF NOT /STATE
0205 349 MOVL WRK C RSLNXT(R10),R8 ;SAVE VALUE TOKEN PTR
0205 350 32$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUE
0205 351 CMPL #PTR_K_COMMA,R4 ;TERMINATOR A COMMA?

```


PC	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op465	
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

```
56 40 AB 7E 02AE 409 60$: MOVAB PRC_Q_KEYPAD(R11),R6 ;GET ADDRESS OF KEYPAD SYMBOL TABLE
5C 56 D0 02B2 410 MOVL R6,AP ;COPY ADDRESS OF TABLE LISTHEAD
02B5 411
02B5 412
02B5 413 : GET NEXT SYMBOL.
02B5 414
56 66 D0 02B5 415 70$: MOVL (R6),R6 ;GET ADDRESS OF NEXT ENTRY
5C 56 D1 02B8 416 CMPL R6,AP ;END OF TABLE?
CE 13 02B8 417 BEQL 58$ ;IF EQL YES
02BD 418
02BD 419
02BD 420 : IF STATE DOES NOT MATCH, THEN SKIP THIS SYMBOL.
02BD 421
54 0C A6 9E 02BD 422 MOVAB SYM_T_SYMBOL(R6),R4 ;GET ADDRESS OF SYMBOL NAME
51 84 9A 02C1 423 MOVZBL (R4)+,R1 ;GET LENGTH OF SYMBOL NAME
54 02 A441 9E 02C4 424 MOVAB 2(R4)(R1),R4 ;GET ADDRESS OF IF_STATE
52 48 AB D0 02C9 425 MOVL PRC_L_CURRKEY(R11),R2 ;GET CURRENT STATE LENGTH/ADDRESS
51 82 9A 02CD 426 MOVZBL (R2)+,R1
84 51 91 02D0 427 CMPB R1,(R4)+ ;STATE LENGTH THE SAME?
E0 12 02D3 428 BNEQ 70$ ;IF DIFF THEN GET NEXT
64 62 51 29 02D5 429 CMPC R1,(R2),(R4) ;STATES MATCH?
DA 12 02D9 430 BNEQ 70$ ;NO, THEN GET NEXT
02DB 431
02DB 432
02DB 433 : STATE DID MATCH. OUTPUT LOG MESSAGE IF REQUESTED. DELETE THE SYMBOL.
02DB 434
53 56 D0 02DB 435 MOVL R6,R3 ;COPY SYMBOL ADDRESS
02 10 02DE 436 BSBB DELKEY ;OUTPUT LOG MESSAGE AND DELETE THE KEY
CC 11 02E0 437 BRB 60$ ;GET NEXT
```

```
02E2 439 :+
02E2 440 : DELKEY - OUTPUT THE DELKEY /LOG MESSAGE AND DELETE THE SPECIFIED KEY.
02E2 441 :-
02E2 442 DELKEY:
02E2 443          BLBC      4(SP),80$          ;OUTPUT DELKEY MSG AND DELETE THE KEY
02E2 444          MOVAB   SYM_T_SYMBOL(R3),-(SP) ;SKIP IF /NOLOG SPECIFIED
02E6 445          PUSHL   PRC_L_CURRKEY(R11)    ;SET ADDR OF ASCII STRING
02EA 446          MOVL    #2,R1                ;SET ADDRESS OF ASCII STATE NAME
02ED 447          MOVL    #CLIS_DELKEY,R0       ;SET FAO COUNT
02F0 448          BSBW    DCL$FORMMSG          ;SET STATUS
02F7 449          BSBW    DCL$DEALLOCSYM       ;OUTPUT THE MESSAGE
02FA 450          RSB      80$:                ;DEALLOCATE KEYPAD ENTRY
02FD                                     ;RETURN
```

14 04 AE E9 02E2 443
7E 0C A3 9E 02E6 444
48 AB DD 02EA 445
51 02 DD 02ED 446
50 0003DDCB 8F DD 02F0 447
FD06' 30 02F7 448
FD03' 30 02FA 449
05 02FD 450

- KEYPAD SYMBOL TABLE MANIPULATION ROUTI 15-SEP-1984 23:59:38 VAX/VMS Macro V04-00
DELETE KEYPAD SYMBOL 4-SEP-1984 23:41:34 [DCL.SRC]KEYPAD.MAR;1

			02FE	452	++	
			02FE	453	UNDKEY - OUTPUT THE UNDKEY WARNING MESSAGE.	
			02FE	454	-	
			02FE	455	UNDKEY:	
7E	56	7D	02FE	456	MOVQ	R6,-(SP)
	5E	DD	0301	457	PUSHL	SP
48	AB	DD	0303	458	PUSHL	PRC_L_CURRKEY(R11)
51	02	DD	0306	459	MOVL	#2,R1
50	00038260	DD	0309	460	MOVL	#CLIS UNDKEY,R0
	FCED	30	0310	461	BSBW	DCL\$FORMMSG
5E	08	CO	0313	462	ADDL	#8,SP
		05	0316	463	RSB	
						:OUTPUT UNDKEY WARNING
						:PUSH DESC OF SYMBOL NAME
						:PUSH DESCR ADDRESS
						:PUSH ADDR OF ASCII STATE
						:SET FAD COUNT
						:SET UNDEFINED SYMBOL STATUS
						:OUTPUT THE MESSAGE
						:RESTORE THE STACK
						:RETURN

```
0317 465 .SBTTL SHOW KEYPAD SYMBOL TABLE ENTRIES
0317 466 :+
0317 467 DCL$SHOWKEY - SHOW KEYPAD SYMBOL TABLE ENTRIES
0317 468 :
0317 469 THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SHOW KEYS
0317 470 COMMAND.
0317 471 :
0317 472 INPUTS:
0317 473 :
0317 474 R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0317 475 R9 = ADDRESS OF SCRATCH STACK.
0317 476 R10 = BASE ADDRESS OF COMMAND WORK AREA.
0317 477 R11 = BASE ADDRESS OF PROCESS WORK AREA.
0317 478 :
0317 479 OUTPUTS:
0317 480 :
0317 481 THE SPECIFIED KEYPAD SYMBOL TABLE ENTRY OR ALL KEYPAD SYMBOL TABLE
0317 482 ENTRIES FOR THE CURRENT OR SPECIFIED STATE ARE WRITTEN TO THE OUTPUT
0317 483 STREAM.
0317 484 :-
0317 485 :
0317 486 DCL$SHOWKEY:: ;SHOW KEYPAD SYMBOL TABLE ENTRIES
0317 487 :
0317 488 INIT PARSE STATE.
0317 489 FLAG BITS ARE: 0 = /ALL, 1 = /BRIEF, 2 = FIRST STATE,
0317 490 3 = UNDKEY, 4 = /DIRECTORY.
0317 491 :
0317 492 PUSHL #1 ;INIT MESSAGE FLAG (ASSUME /LOG)
0317 493 CLRL -(SP) ;CLEAR STATE TOKEN PTR
0317 494 MOVL #2,-(SP) ;INIT FLAGS (ASSUME /ALL /BRIEF)
0317 495 CLRL R6 ;ZERO DESCRIPTOR OF SYMBOL NAME
0317 496 :
0317 497 :
0317 498 : PROCESS THE TOKENS ON THE COMMAND LINE.
0317 499 :
0317 500 10$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUE
0317 501 CMPB #PTR_K_ENDLINE,R5 ;END OF LINE?
0317 502 BEQL 11$ ;BRANCH IF SO
0317 503 CMPB #PTR_K_PARAMETR,R5 ;PARAMETER?
0317 504 BNEQ 12$ ;BRANCH IF QUALIFIER
0317 505 ASSUME PTR_V_KEYWORD EQ 21
0317 506 BBS #1,R3,10$ ;SKIP IF OPTION KEYWORD
0317 507 MOVQ R1,R6 ;SAVE DESCRIPTOR OF SYMBOL NAME
0317 508 BISL #1,(SP) ;INDICATE NOT /ALL
0317 509 BRB 10$ ;GET NEXT TOKEN
0317 510 11$: BRW 20$ ;AT END OF LINE. EXECUTE COMMAND
0317 511 :
0317 512 :
0317 513 : PROCESS QUALIFIERS.
0317 514 :
0317 515 12$: BSBW DCL$GETNVAL ;GET QUALIFIER NUMBER
0317 516 CMPL R1,#CLISK_SHKY_FULL ;/FULL QUALIFIER?
0317 517 BEQL 16$ ;YES, THEN PROCESS
0317 518 CMPL R1,#CLISK_SHKY_BRIE ;/BRIEF QUALIFIER?
0317 519 BEQL 18$ ;YES, THEN PROCESS
0317 520 CMPL R1,#CLISK_SHKY_DIRE ;/DIRECTORY QUALIFIER?
0317 521 BEQL 19$ ;YES, THEN PROCESS
```

01	DD	0317	492	PUSHL	#1	;INIT MESSAGE FLAG (ASSUME /LOG)	
7E	D4	0319	493	CLRL	-(SP)	;CLEAR STATE TOKEN PTR	
02	D0	031B	494	MOVL	#2,-(SP)	;INIT FLAGS (ASSUME /ALL /BRIEF)	
56	7C	031E	495	CLRL	R6	;ZERO DESCRIPTOR OF SYMBOL NAME	
		0320	496				
		0320	497				
		0320	498				
		0320	499				
FCDD'	30	0320	500	10\$: BSBW	DCL\$GETDVAL	;GET NEXT DESCRIPTOR VALUE	
55	04	91	0323	501	CMPB #PTR_K_ENDLINE,R5	;END OF LINE?	
	11	13	0326	502	BEQL 11\$;BRANCH IF SO	
55	03	91	0328	503	CMPB #PTR_K_PARAMETR,R5	;PARAMETER?	
	0F	12	032B	504	BNEQ 12\$;BRANCH IF QUALIFIER	
			032D	505	ASSUME PTR_V_KEYWORD EQ 21		
EF	53	01	E0	032D	506	BBS #1,R3,10\$;SKIP IF OPTION KEYWORD
	56	51	7D	0331	507	MOVQ R1,R6	;SAVE DESCRIPTOR OF SYMBOL NAME
	6E	01	C8	0334	508	BISL #1,(SP)	;INDICATE NOT /ALL
		E7	11	0337	509	BRB 10\$;GET NEXT TOKEN
	007A	31	0339	510	11\$: BRW	20\$;AT END OF LINE. EXECUTE COMMAND
			033C	511			
			033C	512			
			033C	513			
			033C	514			
FCC1'	30	033C	515	12\$: BSBW	DCL\$GETNVAL	;GET QUALIFIER NUMBER	
00000000'8F	51	D1	033F	516	CMPL R1,#CLISK_SHKY_FULL	; /FULL QUALIFIER?	
	48	13	0346	517	BEQL 16\$;YES, THEN PROCESS	
00000000'8F	51	D1	0348	518	CMPL R1,#CLISK_SHKY_BRIE	; /BRIEF QUALIFIER?	
	48	13	034F	519	BEQL 18\$;YES, THEN PROCESS	
00000000'8F	51	D1	0351	520	CMPL R1,#CLISK_SHKY_DIRE	; /DIRECTORY QUALIFIER?	
	4F	13	0358	521	BEQL 19\$;YES, THEN PROCESS	

```
00000000'8F 51 D1 035A 522 CMPL R1,#CLISK_SHKY_LOG ;/LOG QUALIFIER?
1F 13 0361 523 BEQL 14$ ;YES, THEN PROCESS
00000000'8F 51 D1 0363 524 CMPL R1,#CLISK_SHKY_STAT ;/STATE QUALIFIER?
B4 12 036A 525 BNEQ 10$ ;NO, THEN IGNORE
036C 526
036C 527
036C 528
036C 529
036C 530
04 AE 04 AE D4 036C 530 CLRL 4(SP) ;ASSUME /NOSTATE
AD 53 00 E0 036F 531 BBS #PTR_V NEGATE-PTR_V_FLAGS,R3,10$ ;IGNORE IF NOT /STATE
04 AE BA AA D0 0373 532 MOVL WRK [RSLNXT(R10),4(SP)] ;SAVE VALUE TOKEN PTR
FC85' 30 0378 533 13$: BSBW DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUE
54 05 D1 037B 534 CMPL #PTR_K_COMMA,R4 ;TERMINATOR A COMMA?
F8 13 037E 535 BEQL 13$ ;GET NEXT VALUE
9E 11 0380 536 BRB 10$ ;GET NEXT
0382 537
0382 538
0382 539
0382 540
0382 541
08 AE 01 C8 0382 541 14$: BISL #1,8(SP) ;ASSUME /LOG
96 53 00 E1 0386 542 BBC #PTR_V NEGATE-PTR_V_FLAGS,R3,10$ ;IGNORE IF /NOLOG
08 AE 01 CA 038A 543 BICL #1,8(SP) ;SET FLAG TO /NOLOG
90 11 038E 544 BRB 10$ ;GET NEXT TOKEN
0390 545
0390 546
0390 547
0390 548
0393 549
0397 550
039A 551
039C 552
039F 553
03A3 554
03A6 555
03A9 556
03AC 557
03B0 558
03B3 559
03B6 560
03B6 561
03B6 562
03B6 563
03B6 564
3E 6E 04 E1 03B6 564 20$: BBC #4,(SP),21$ ;BRANCH IF /NODIRECTORY
5C 40 AB 7E 03BA 565 MOVAQ PRG_Q_KEYPAD(R11),AP ;GET ADDRESS OF KEYPAD SYMBOL TABLE
5C 56 7C 03C0 566 PUSHL AP ;SAVE R6
5C 6C D0 03C2 567 CLRQ R6 ;SET INITIAL STATE DESCRIPTOR
6E 5C D1 03C5 568 210$: MOVL (AP),AP ;GET ADDRESS OF NEXT ENTRY
2A 13 03C8 569 CMPL AP,(SP) ;END OF TABLE?
DC AC 9A 03CA 570 BEQL 230$ ;IF EQL, THEN DONE
OF AC 41 9E 03CE 571 MOVZBL SYM_T_SYMBOL(AP),R1 ;GET LENGTH OF SYMBOL
51 51 82 9A 03D3 572 MOVAB SYM_T_SYMBOL+3(AP)[R1],R2 ;GET ADDRESS OF IF STATE LENGTH
56 51 D1 03D6 573 MOVZBL (R2)+,R1 ;GET IF STATE LENGTH
11 12 03D9 574 CMPL R1,R6 ;STATES MATCH
7E 51 7D 03DB 575 BNEQ 220$ ;NO, LIST IT
67 62 51 29 03DE 576 MOVQ R1,-(SP) ;SAVE STATE DESCRIPTOR
05 12 03E2 577 CMPC3 R1,(R2),(R7) ;STATES MATCH?
BNEQ 215$ ;NO, THEN OUTPUT THE STATE
```



```
51 8E 7D 03E4 579      MOVQ    (SP)+,R1      ;RESTORE STATE DESCRIPTOR
      D9 11 03E7 580      BRB      210$      ;YES, THEN SKIP
51 8E 7D 03E9 581 215$:  MOVQ    (SP)+,R1      ;RESTORE STATE DESCRIPTOR
56 51 7D 03EC 582 220$:  MOVQ    R1,R6      ;SAVE NEW STATE
      FCOE' 30 03EF 583      BSBW    DCL$MSGOUT  ;OUTPUT THE STATE
      CE 11 03F2 584      BRB      210$      ;GET NEXT STATE
      8E D5 03F4 585 230$:  TSTL    (SP)+      ;RESTORE THE STACK
      49 11 03F6 586      BRB      90$      ;EXIT
      03F8 587
      03F8 588
      03F8 589
      03F8 590
      03F8 591
BA AA 04 AE D0 03F8 592 21$:  MOVL    4(SP),WRK_L_RSLNXT(R10) ;RESET FOR FIRST STATE VALUE
      OA 13 03FD 593      BEQL    23$      ;SKIP IF NONE
      FBFE' 30 03FF 594 22$:  BSBW    DCL$GETDVAL ;GET NEXT DESCRIPTOR VALUE
04 AE 54 D0 0402 595      MOVL    R4,4(SP) ;SAVE THE TERMINATOR
      038B 30 0406 596      BSBW    DCL$ALLOC_STATE ;SET NEW STATE
      0409 597
      0409 598
      0409 599
      0409 600 23$:  BLBC    (SP),40$      ;BRANCH IF /ALL
      040C 601
      040C 602
      040C 603
      040C 604
51 56 7D 040C 605      MOVQ    R6,R1      ;GET DESCRIPTOR OF SYMBOL NAME
50 08 AE D0 040F 606      MOVL    8(SP),R0 ;GET /LOG FLAG
      023B 30 0413 607      BSBW    DCL$SYNONYM ;CHECK FOR SYNONYM KEY NAMES
56 51 7D 0416 608      MOVQ    R1,R6      ;SAVE DESCRIPTOR IN CASE OF UNDKEY
      02C2 30 0419 609      BSBW    DCL$FIND_KEYPAD ;FIND SPECIFIED SYMBOL
      11 50 E8 041C 610      BLBS    R0,35$ ;BRANCH IF FOUND
      041F 611
      041F 612
      041F 613
      041F 614
05 6E 02 E3 041F 615      BBCS    #2,(SP),33$ ;SKIP BLANK LINE IF FIRST LINE
      51 7C 0423 616      CLRQ    R1      ;SET NULL STRING
      FBDB' 30 0425 617      BSBW    DCL$MSGOUT ;OUTPUT THE BLANK LINE
6E 08 C8 0428 618 33$:  BISL    #8,(SP) ;SET UNDEFINED SYMBOL FLAG
      FED0 30 042B 619      BSBW    UNDKEY ;OUTPUT UNDEFINED KEY MSG
      06 11 042E 620      BRB      38$      ;GET NEXT TABLE
      0430 621
      0430 622
      0430 623
      0430 624
      0430 625 35$:  BSBW    DISPHDR ;DISPLAY KEYPAD TABLE NAME
      0060 30 0433 626      BSBW    DISPSYMB ;DISPLAY THE SYMBOL DATA
04 AE 05 D1 0436 627 38$:  CMPL    #PTR_K_COMMA,4(SP) ;TERMINATOR A COMMA?
      05 12 043A 628      BNEQ    90$      ;NO, TIME TO EXIT
      FBC1' 30 043C 629      BSBW    DCL$LOCKED_STATE ;YES, RESTORE LOCKED KEY STATE
      BE 11 043F 630      BRB      22$      ;BEFORE GETTING NEXT STATE
      0441 631
      0441 632
      0441 633
      0441 634
      0441 635
      0441 636
      0441 637
      0441 638
      0441 639
      0441 640
      0441 641
      0441 642
      0441 643
      0441 644
      0441 645
      0441 646
      0441 647
      0441 648
      0441 649
      0441 650
      0441 651
      0441 652
      0441 653
      0441 654
      0441 655
      0441 656
      0441 657
      0441 658
      0441 659
      0441 660
      0441 661
      0441 662
      0441 663
      0441 664
      0441 665
      0441 666
      0441 667
      0441 668
      0441 669
      0441 670
      0441 671
      0441 672
      0441 673
      0441 674
      0441 675
      0441 676
      0441 677
      0441 678
      0441 679
      0441 680
      0441 681
      0441 682
      0441 683
      0441 684
      0441 685
      0441 686
      0441 687
      0441 688
      0441 689
      0441 690
      0441 691
      0441 692
      0441 693
      0441 694
      0441 695
      0441 696
      0441 697
      0441 698
      0441 699
      0441 700
      0441 701
      0441 702
      0441 703
      0441 704
      0441 705
      0441 706
      0441 707
      0441 708
      0441 709
      0441 710
      0441 711
      0441 712
      0441 713
      0441 714
      0441 715
      0441 716
      0441 717
      0441 718
      0441 719
      0441 720
      0441 721
      0441 722
      0441 723
      0441 724
      0441 725
      0441 726
      0441 727
      0441 728
      0441 729
      0441 730
      0441 731
      0441 732
      0441 733
      0441 734
      0441 735
      0441 736
      0441 737
      0441 738
      0441 739
      0441 740
      0441 741
      0441 742
      0441 743
      0441 744
      0441 745
      0441 746
      0441 747
      0441 748
      0441 749
      0441 750
      0441 751
      0441 752
      0441 753
      0441 754
      0441 755
      0441 756
      0441 757
      0441 758
      0441 759
      0441 760
      0441 761
      0441 762
      0441 763
      0441 764
      0441 765
      0441 766
      0441 767
      0441 768
      0441 769
      0441 770
      0441 771
      0441 772
      0441 773
      0441 774
      0441 775
      0441 776
      0441 777
      0441 778
      0441 779
      0441 780
      0441 781
      0441 782
      0441 783
      0441 784
      0441 785
      0441 786
      0441 787
      0441 788
      0441 789
      0441 790
      0441 791
      0441 792
      0441 793
      0441 794
      0441 795
      0441 796
      0441 797
      0441 798
      0441 799
      0441 800
      0441 801
      0441 802
      0441 803
      0441 804
      0441 805
      0441 806
      0441 807
      0441 808
      0441 809
      0441 810
      0441 811
      0441 812
      0441 813
      0441 814
      0441 815
      0441 816
      0441 817
      0441 818
      0441 819
      0441 820
      0441 821
      0441 822
      0441 823
      0441 824
      0441 825
      0441 826
      0441 827
      0441 828
      0441 829
      0441 830
      0441 831
      0441 832
      0441 833
      0441 834
      0441 835
      0441 836
      0441 837
      0441 838
      0441 839
      0441 840
      0441 841
      0441 842
      0441 843
      0441 844
      0441 845
      0441 846
      0441 847
      0441 848
      0441 849
      0441 850
      0441 851
      0441 852
      0441 853
      0441 854
      0441 855
      0441 856
      0441 857
      0441 858
      0441 859
      0441 860
      0441 861
      0441 862
      0441 863
      0441 864
      0441 865
      0441 866
      0441 867
      0441 868
      0441 869
      0441 870
      0441 871
      0441 872
      0441 873
      0441 874
      0441 875
      0441 876
      0441 877
      0441 878
      0441 879
      0441 880
      0441 881
      0441 882
      0441 883
      0441 884
      0441 885
      0441 886
      0441 887
      0441 888
      0441 889
      0441 890
      0441 891
      0441 892
      0441 893
      0441 894
      0441 895
      0441 896
      0441 897
      0441 898
      0441 899
      0441 900
      0441 901
      0441 902
      0441 903
      0441 904
      0441 905
      0441 906
      0441 907
      0441 908
      0441 909
      0441 910
      0441 911
      0441 912
      0441 913
      0441 914
      0441 915
      0441 916
      0441 917
      0441 918
      0441 919
      0441 920
      0441 921
      0441 922
      0441 923
      0441 924
      0441 925
      0441 926
      0441 927
      0441 928
      0441 929
      0441 930
      0441 931
      0441 932
      0441 933
      0441 934
      0441 935
      0441 936
      0441 937
      0441 938
      0441 939
      0441 940
      0441 941
      0441 942
      0441 943
      0441 944
      0441 945
      0441 946
      0441 947
      0441 948
      0441 949
      0441 950
      0441 951
      0441 952
      0441 953
      0441 954
      0441 955
      0441 956
      0441 957
      0441 958
      0441 959
      0441 960
      0441 961
      0441 962
      0441 963
      0441 964
      0441 965
      0441 966
      0441 967
      0441 968
      0441 969
      0441 970
      0441 971
      0441 972
      0441 973
      0441 974
      0441 975
      0441 976
      0441 977
      0441 978
      0441 979
      0441 980
      0441 981
      0441 982
      0441 983
      0441 984
      0441 985
      0441 986
      0441 987
      0441 988
      0441 989
      0441 990
      0441 991
      0441 992
      0441 993
      0441 994
      0441 995
      0441 996
      0441 997
      0441 998
      0441 999
      0441 1000
      0441 1001
      0441 1002
      0441 1003
      0441 1004
      0441 1005
      0441 1006
      0441 1007
      0441 1008
      0441 1009
      0441 1010
      0441 1011
      0441 1012
      0441 1013
      0441 1014
      0441 1015
      0441 1016
      0441 1017
      0441 1018
      0441 1019
      0441 1020
      0441 1021
      0441 1022
      0441 1023
      0441 1024
      0441 1025
      0441 1026
      0441 1027
      0441 1028
      0441 1029
      0441 1030
      0441 1031
      0441 1032
      0441 1033
      0441 1034
      0441 1035
      0441 1036
      0441 1037
      0441 1038
      0441 1039
      0441 1040
      0441 1041
      0441 1042
      0441 1043
      0441 1044
      0441 1045
      0441 1046
      0441 1047
      0441 1048
      0441 1049
      0441 1050
      0441 1051
      0441 1052
      0441 1053
      0441 1054
      0441 1055
      0441 1056
      0441 1057
      0441 1058
      0441 1059
      0441 1060
      0441 1061
      0441 1062
      0441 1063
      0441 1064
      0441 1065
      0441 1066
      0441 1067
      0441 1068
      0441 1069
      0441 1070
      0441 1071
      0441 1072
      0441 1073
      0441 1074
      0441 1075
      0441 1076
      0441 1077
      0441 1078
      0441 1079
      0441 1080
      0441 1081
      0441 1082
      0441 1083
      0441 1084
      0441 1085
      0441 1086
      0441 1087
      0441 1088
      0441 1089
      0441 1090
      0441 1091
      0441 1092
      0441 1093
      0441 1094
      0441 1095
      0441 1096
      0441 1097
      0441 1098
      0441 1099
      0441 1100
      0441 1101
      0441 1102
      0441 1103
      0441 1104
      0441 1105
      0441 1106
      0441 1107
      0441 1108
      0441 1109
      0441 1110
      0441 1111
      0441 1112
      0441 1113
      0441 1114
      0441 1115
      0441 1116
      0441 1117
      0441 1118
      0441 1119
      0441 1120
      0441 1121
      0441 1122
      0441 1123
      0441 1124
      0441 1125
      0441 1126
      0441 1127
      0441 1128
      0441 1129
      0441 1130
      0441 1131
      0441 1132
      0441 1133
      0441 1134
      0441 1135
      0441 1136
      0441 1137
      0441 1138
      0441 1139
      0441 1140
      0441 1141
      0441 1142
      0441 1143
      0441 1144
      0441 1145
      0441 1146
      0441 1147
      0441 1148
      0441 1149
      0441 1150
      0441 1151
      0441 1152
      0441 1153
      0441 1154
      0441 1155
      0441 1156
      0441 1157
      0441 1158
      0441 1159
      0441 1160
      0441 1161
      0441 1162
      0441 1163
      0441 1164
      0441 1165
      0441 1166
      0441 1167
      0441 1168
      0441 1169
      0441 1170
      0441 1171
      0441 1172
      0441 1173
      0441 1174
      0441 1175
      0441 1176
      0441 1177
      0441 1178
      0441 1179
      0441 1180
      0441 1181
      0441 1182
      0441 1183
      0441 1184
      0441 1185
      0441 1186
      0441 1187
      0441 1188
      0441 1189
      0441 1190
      0441 1191
      0441 1192
      0441 1193
      0441 1194
      0441 1195
      0441 1196
      0441 1197
      0441 1198
      0441 1199
      0441 1200
      0441 1201
      0441 1202
      0441 1203
      0441 1204
      0441 1205
      0441 1206
      0441 1207
      0441 1208
      0441 1209
      0441 1210
      0441 1211
      0441 1212
      0441 1213
      0441 1214
      0441 1215
      0441 1216
      0441 1217
      0441 1218
      0441 1219
      0441 1220
      0441 1221
      0441 1222
      0441 1223
      0441 1224
      0441 1225
      0441 1226
      0441 1227
      0441 1228
      0441 1229
      0441 1230
      0441 1231
      0441 1232
      0441 1233
      0441 1234
      0441 1235
      0441 1236
      0441 1237
      0441 1238
      0441 1239
      0441 1240
      0441 1241
      0441 1242
      0441 1243
      0441 1244
      0441 1245
      0441 1246
      0441 1247
      0441 1248
      0441 1249
      0441 1250
      0441 1251
      0441 1252
      0441 1253
      0441 1254
      0441 1255
      0441 1256
      0441 1257
      0441 1258
      0441 1259
      0441 1260
      0441 1261
      0441 1262
      0441 1263
      0441 1264
      0441 1265
      0441 1266
      0441 1267
      0441 1268
      0441 1269
      0441 1270
      0441 1271
      0441 1272
      0441 1273
      0441 1274
      0441 1275
      0441 1276
      0441 1277
      0441 1278
      0441 1279
      0441 1280
      0441 1281
      0441 1282
      0441 1283
      0441 1284
      0441 1285
      0441 1286
      0441 1287
      0441 1288
      0441 1289
      0441 1290
      0441 1291
      0441 1292
      0441 1293
      0441 1294
      0441 1295
      0441 1296
      0441 1297
      0441 1298
      0441 1299
      0441 1300
      0441 1301
      0441 1302
      0441 1303
      0441 1304
      0441 1305
      0441 1306
      0441 1307
      0441 1308
      0441 1309
      0441 1310
      0441 1311
      0441 1312
      0441 1313
      0441 1314
      0441 1315
      0441 1316
      0441 1317
      0441 1318
      0441 1319
      0441 1320
      0441 1321
      0441 1322
      0441 1323
      0441 1324
      0441 1325
      0441 1326
      0441 1327
      0441 1328
      0441 1329
      0441 1330
      0441 1331
      0441 1332
      0441 1333
      0441 1334
      0441 1335
      0441 1336
      0441 1337
      0441 1338
      0441 1339
      0441 1340
      0441 1341
      0441 1342
      0441 1343
      0441 1344
      0441 1345
      0441 1346
      0441 1347
      0441 1348
      0441 1349
      0441 1350
      0441 1351
      0441 1352
      0441 1353
      0441 1354
      0441 1355
      0441 1356
      0441 1357
      0441 1358
      0441 1359
      0441 1360
      0441 1361
      0441 1362
      0441 1363
      0441 1364
      0441 1365
      0441 1366
      0441 1367
      0441 1368
      0441 1369
      0441 1370
      0441 1371
      0441 1372
      0441 1373
      0441 1374
      0441 1375
      0441 1376
      0441 1377
      0441 1378
      0441 1379
      0441 1380
      0441 1381
      0441 1382
      0441 1383
      0441 1384
      0441 1385
      0441 1386
      0441 1387
      0441 1388
      0441 1389
      0441 1390
      0441 1391
      0441 1392
      0441 1393
      0441 1394
      0441 1395
      0441 1396
      0441 1397
      0441 1398
      0441 1399
      0441 1400
      0441 1401
      0441 1402
      0441 1403
      0441 1404
      0441 1405
      0441 1406
      0441 1407
      0441 1408
      0441 1409
      0441 1410
      0441 1411
      0441 1412
      0441 1413
      0441 1414
      0441 1415
      0441 1416
      0441 1417
      0441 1418
      0441 1419
      0441 1420
      0441 1421
      0441 1422
      0441 1423
      0441 1424
      0441 1425
      0441 1426
      0441 1427
      0441 1428
      0441 1429
      0441 1430
      0441 1431
      0441 1432
      0441 1433
      0441 1434
      0441 1435
      0441 1436
      0441 1437
      0441 1438
      0441 1439
      0441 1440
      0441 1441
      0441 1442
      0441 1443
      0441 1444
      0441 1445
      0441 1446
      0441 1447
      0441 1448
      0441 1449
      0441 1450
      0441 1451
      0441 1452
      0441 1453
      0441 1454
      0441 1455
      0441 1456
      0441 1457
      0441 1458
      0441 1459
      0441 1460
      0441 1461
      0441 1462
      0441 1463
      0441 1464
      0441 1465
      0441 1466
      0441 1467
      0441 1468
      0441 1469
      0441 1470
      0441 1471
      0441 1472
      0441 1473
      0441 1474
      0441 1475
      0441 1476
      0441 1477
      0441 1478
      0441 1479
      0441 1480
      0441 1481
      0441 1482
      0441 1483
      0441 1484
      0441 1485
      0441 1486
      0441 1487
      0441 1488
      0441 1489
      0441 1490
      0441 1491
      0441 1492
      0441 1493
      0441 1494
      0441 1495
      0441 1496
      0441 1497
      0441 1498
      0441 1499
      0441 1500
      0441 1501
      0441 1502
      0441 1503
      0441 1504
      0441 1505
      0441 1506
      0441 1507
      0441 1508
      0441 1509
      0441 1510
      0441 1511
      0441 1512
      0441 1513
      0441 1514
      0441 1515
      0441 1516
      0441 1517
      0441 1518
      0441 1519
      0441 1520
      0441 1521
      0441 1522
      0441 1523
      0441 1524
      0441 1525
      0441 1526
      0441 1527
      0441 1528
      0441 1529
      0441 1530
      0441 1531
      0441 1532
      0441 1533
      0441 1534
      0441 1535
      0441 1536
      0441 1537
      0441 1538
      0441 1539
      0441 1540
      0441 1541
      0441 1542
      0441 1543
      0441 1544
      0441 1545
      0441 1546
      0441 1547
      0441 1548
      0441 1549
      0441 1550
      0441 1551
      0441 1552
      0441 1553
      0441 1554
      0441 1555
      0441 1556
      0441 1557
      0441 1558
      0441 1559
      0441 1560
      0441 1561
      0441 1562
      0441 15
```

```
50 07 51 8E DO 0441 636 90$: STATUS NORMAL ;ASSUME SUCCESSFUL COMPLETION
    10038260 03 E1 0448 637 MOVL (SP)+,R1 ;GET FLAGS
    SE 08 CO 044B 638 BBC #3,R1,95$ ;BRANCH IF NO UNDEFINED SYMBOLS
    FBA4 05 044F 639 MOVL #CLIS_UNDKY!STSM_INHIB_MSG,R0 ;SET STATUS, INHIBIT RESIGNAL
    30 0456 640 95$: ADDL #8,SP ;RESTORE THE STACK
    05 0459 641 BSBW DCL$LOCKED_STATE ;RESTORE KEY STATE
    045C 642 RSB
    045D 643
    045D 644 ;
    045D 645 ; DISPLAY ALL SYMBOL ENTRIES FOR THE SPECIFIED OR CURRENT STATE.
    045D 646
    56 0123 30 045D 647 40$: BSBW DISPHDR ;DISPLAY KEYPAD TABLE NAME
    40 AB 7E 0460 648 MOVAQ PRC_Q_KEYPAD(R11),R6 ;GET ADDRESS OF KEYPAD SYMBOL TABLE
    SC 56 DO 0464 649 MOVL R6,AP ;COPY ADDRESS OF TABLE LISTHEAD
    0467 650
    0467 651 ;
    0467 652 ; GET NEXT SYMBOL.
    0467 653
    56 66 DO 0467 654 50$: MOVL (R6),R6 ;GET ADDRESS OF NEXT ENTRY
    SC 56 D1 046A 655 CMPL R6,AP ;END OF TABLE?
    C7 13 046D 656 BEQL 38$ ;IF EQL YES
    046F 657
    046F 658 ;
    046F 659 ; IF STATE DOES NOT MATCH, THEN SKIP THIS SYMBOL.
    046F 660
    54 0C A6 9E 046F 661 MOVAB SYM_T_SYMBOL(R6),R4 ;GET ADDRESS OF SYMBOL NAME
    51 84 9A 0473 662 MOVZBL (R4)+,R1 ;GET LENGTH OF SYMBOL NAME
    54 02 A441 9E 0476 663 MOVAB 2(R4)[R1],R4 ;GET ADDRESS OF IF STATE
    50 84 9A 047B 664 MOVZBL (R4)+,R0 ;GET IF STATE LENGTH
    52 48 AB DO 047E 665 MOVL PRC_L_CURRKEY(R11),R2 ;GET CURRENT STATE LENGTH/ADDRESS
    51 82 9A 0482 666 MOVZBL (R2)+,R1
    64 50 00 62 51 2D 0485 667 CMPC5 R1,(R2),#0,R0,(R4) ;STATES MATCH?
    A9 19 0488 668 65$: BLSS 38$ ;NO, GET NEXT STATE
    D8 14 048D 669 BGTR 50$ ;NO, GET NEXT SYMBOL
    048F 670
    048F 671 ;
    048F 672 ; STATE DID MATCH. DISPLAY THE SYMBOL.
    048F 673
    53 56 DO 048F 674 70$: MOVL R6,R3 ;SET ADDRESS OF SYMBOL
    02 10 0492 675 BSBB DISPSYMB ;FORMAT AND OUTPUT ENTRY
    D1 11 0494 676 BRB 50$ ;GET NEXT
```

```
0496 678 :+
0496 679 : DISPSYMB - DISPLAY THE VALUE AND ATTRIBUTES OF A GIVEN KEYPAD SYMBOL.
0496 680 :
0496 681 : INPUTS:
0496 682 :
0496 683 : 4(SP) = FLAGS LONGWORD - BIT 1 IS SET IF /BRIEF
0496 684 : R3 = ADDRESS OF SYMBOL TABLE ENTRY
0496 685 : R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0496 686 : R9 = ADDRESS OF SCRATCH STACK.
0496 687 :
0496 688 : DISPSYMB:
50 04 AE DO 0496 689 : MOVL 4(SP),R0 :FORMAT A SYMBOL
02C0 8F BB 049A 690 : PUSH R3 :GET THE FLAGS
68 DD 049E 691 : PUSH R8 :SAVE REGISTERS
58 50 DO 04A0 692 : PUSH R8 :SAVE SCRATCH DESCR LENGTH
04A2 693 : MOV R8,R8 :SAVE ADDR OF SCRATCH DESCR
04A5 694 : :COPY THE FLAGS
04A5 695 :
04A5 696 : GET AND SAVE DESCRIPTOR OF SYMBOL NAME.
52 0C A3 9E 04A5 697 :
51 82 9A 04A9 698 : MOVAB SYM_T_SYMBOL(R3),R2 :POINT TO SYMBOL NAME
79 51 7D 04AC 699 : MOVZBL (R2)+,R1 :GET NAME LENGTH
57 59 DO 04AF 700 : MOVQ R1,-(R9) :BUILD NAME DESCRIPTOR
04B2 701 : MOVL R9,R7 :COPY SCRATCH STACK POINTER
04B2 702 :
04B2 703 : GET AND SAVE DESCRIPTOR OF SYMBOL VALUE.
52 02 A241 9E 04B2 704 :
51 82 9A 04B7 705 : MOVAB 2(R2)[R1],R2 :GET ADDRESS OF IF STATE LENGTH
52 6241 9E 04BA 706 : MOVZBL (R2)+,R1 :GET LENGTH OF IF STATE
51 82 9E 04BE 707 : MOVAB (R2)[R1],R2 :GET ADDRESS OF SYMBOL VALUE LENGTH
79 51 3C 04BE 708 : MOVZBL (R2)+,R1 :GET LENGTH/ADDRESS OF VALUE
56 59 DO 04C1 709 : MOVQ R1,-(R9) :SAVE VALUE DESCRIPTOR
04C4 710 : MOVL R9,R6 :COPY SCRATCH STACK POINTER
04C7 711 :
04C7 712 : GET AND SAVE DESCRIPTOR OF SET_STATE STRING.
04C7 713 :
04C7 714 :
52 6241 9E 04C7 715 : MOVAB (R2)[R1],R2 :GET ADDRESS OF SET_STATE LENGTH
51 82 9A 04CB 716 : MOVZBL (R2)+,R1 :GET LENGTH/ADDRESS OF STATE
02 58 01 E1 04CE 717 : BBC #1,R8,10$ :SKIP IF /NOBRIEF
79 51 D4 04D2 718 : CLRL R1 :PUSH NULL STRING
55 59 DO 04D4 719 : MOVQ R1,-(R9) :SAVE STATE DESCRIPTOR
04D7 720 : MOVL R9,R5 :COPY SCRATCH STACK POINTER
04DA 721 :
04DA 722 : CREATE AND SAVE DESCRIPTOR OF ASCII FAO STRING. OUTPUT WILL LOOK LIKE:
04DA 723 :
04DA 724 : symbol = "value" (ECHO,TERMINATE,ERASE,LOCK,STATE=state)
04DA 725 :
04DA 726 :
52 FB48 CF 9E 04DA 727 :
05 58 01 E1 04DF 728 : MOVAB FULLFAO,R2 :ASSUME FULL DISPLAY
52 FB31 CF 9E 04E3 729 : BBC #1,R8,20$ :SKIP IF /NOBRIEF
51 82 9A 04E8 730 : MOVAB BRIEFAO,R2 :SET BRIEF DISPLAY
79 51 7D 04EB 731 : MOVZBL (R2)+,R1 :MAKE INTO DESCRIPTOR
54 59 DO 04EE 732 : MOVQ R1,-(R9) :AND PUSH ONTO STACK
21 58 01 E0 04F1 733 : MOVL R9,R4 :COPY SCRATCH STACK POINTER
04F1 734 : BBS #1,R8,30$ :SKIP IF /BRIEF
```



```
04F5 735
04F5 736
04F5 737 :: CREATE FAO PARAMETER LIST. ASSUME NO ATTRIBUTES.
04F5 738 ::
79 79 55 DO 04F5 739
79 FB74 CF 9E 04F8 740
79 FB6F CF 9E 04FD 741
79 FB6A CF 9E 0502 742
79 FB65 CF 9E 0507 743
79 FB60 CF 9E 050C 744
79 FB5B CF 9E 0511 745
79 79 56 DO 0516 746 30$:
79 79 57 DO 0519 747
57 59 DO 051C 748
051F 749
051F 750
051F 751 :: NOW RESET FAO ARGUMENTS FOR ANY ATTRIBUTES THAT WERE ABSENT.
051F 752 ::
03 58 01 E1 051F 753
003D 31 0523 754
0526 755
0526 756 40$:
06 0B A3 E0 0528 757
FB37 CF 9E 052B 758
0531 759
0531 760 50$:
06 0B A3 E0 0531 761
FB2C CF 9E 0533 762
0536 763
053C 764 60$:
06 0B A3 E0 053C 765
FB21 CF 9E 053E 766
0541 767
0547 768 70$:
06 0B A3 E0 0547 769
FB16 CF 9E 0549 770
054C 771
0552 772 80$:
0C 0B A3 E1 0552 773
FB16 CF 9E 0554 774
FB08 CF 9E 0557 775
055D 776
0563 777
0563 778 :: FORMAT AND OUTPUT THE MESSAGE
0563 779
58 8ED0 0563 780 90$:
51 68 7D 0566 781
FA85' 30 0575 782
68 8ED0 0578 783
02C0 8F BA 057B 784
057E 785
0582 786
04F5 735
04F5 736
04F5 737 :: CREATE FAO PARAMETER LIST. ASSUME NO ATTRIBUTES.
04F5 738 ::
MOV R5, -(R9) ;SET ADDR OF STATE DESCR
MOVAB NULL, -(R9) ;ASSUME STATE FLAG NOT SET
MOVAB NULL, -(R9) ;ASSUME STATE FLAG NOT SET
MOVAB NULL, -(R9) ;ASSUME LOCK FLAG SET
MOVAB NULL, -(R9) ;ASSUME ERASE FLAG SET
MOVAB NULL, -(R9) ;ASSUME TERMINATE FLAG SET
MOVAB NULL, -(R9) ;ASSUME ECHO FLAG SET
MOV R6, -(R9) ;SET ADDR OF VALUE DESCR
MOV R7, -(R9) ;SET ADDR OF NAME DESCR
MOV R9, R7 ;SAVE ADDRESS OF PARAMETER LIST

:: NOW RESET FAO ARGUMENTS FOR ANY ATTRIBUTES THAT WERE ABSENT.
BBC #1, R8, 40$ ;SKIP IF /NOBRIEF
BRW 90$ ;BRANCH IF /BRIEF
40$: BBS #SYM V ECHO, - ;IS ECHO SET?
SYM B FLAGS(R3), 50$
MOVAB NO, 8(R9)
50$: BBS #SYM V TERMINATE, - ;IS TERMINATE SET?
SYM B FLAGS(R3), 60$
MOVAB NO, T2(R9)
60$: BBS #SYM V ERASE, - ;IS ERASE SET?
SYM B FLAGS(R3), 70$
MOVAB NO, T6(R9)
70$: BBS #SYM V LOCK, - ;IS LOCK SET?
SYM B FLAGS(R3), 80$
MOVAB NO, 20(R9)
80$: BBC #SYM V STATE, - ;IS STATE SET?
SYM B FLAGS(R3), 90$
MOVAB COMMA, 24(R9)
MOVAB STATE, 28(R9)

90$: POPL R8 ;RESTORE SCRATCH DESCRIPTOR
$FAOL_S (R4), (R8), (R8), (R7) ;FORMAT OUTPUT MESSAGE
MOVQ (R8), R1 ;GET OUTPUT MESSAGE PARAMETERS
BSBW DCL$MSGOUT ;OUTPUT THE MESSAGE
POPL (R8) ;RESTORE SCRATCH DESCR LENGTH
POPR #^M<R6, R7, R9> ;RESTORE REGISTERS
RSB ;RETURN
```

```
0583 788 :+
0583 789 :DISPHDR - DISPLAY A KEYPAD TABLE HEADER
0583 790 :
0583 791 :INPUTS:
0583 792 :
0583 793 :4(SP) = FLAGS LONGWORD - BIT 2 IS CLEAR IF FIRST TABLE
0583 794 :R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0583 795 :R9 = ADDRESS OF SCRATCH STACK.
0583 796 :PRC_L_CURRKEY(R11) = ADDRESS OF ASCIC TABLE NAME
0583 797 :-
0583 798 DISPHDR:
0583 799 BBCS #2,4(SP),10$ :DISPLAY KEYPAD TABLE HEADER
0588 800 CLRQ R1 :SKIP BLANK LINE IF FIRST HEADER
058A 801 BSBW DCL$MSGOUT :SET NULL STRING
058D 802 10$: PUSHL (R8) :OUTPUT THE BLANK LINE
058F 803 MOVAB SHOWHDR,R2 :SAVE BUFFER SIZE
0594 804 MOVZBL (R2)+,R1 :GET ADDRESS OF ASCIC FAO STRING
0597 805 MOVQ R1,-(SP) :MAKE INTO DESCRIPTOR
059A 806 MOVL SP,R0 :AND PUSH ONTO STACK
059D 807 MOVAB PRC_L_CURRKEY(R11),R1 :SAVE ITS ADDRESS
05A1 808 $FAO_S (R0),R8),(R8),(R1) :GET ADDRESS OF ASCIC STATE
05B0 809 MOVQ (R8),R1 :FORMAT OUTPUT MESSAGE
05B3 810 BSBW DCL$MSGOUT :GET OUTPUT MESSAGE PARAMETERS
05B6 811 ADDL #8,SP :OUTPUT THE MESSAGE
05B9 812 MOVL (SP)+,(R8) :RESTORE THE STACK
05BC 813 RSB :RESTORE BUFFER SIZE
:RETURN
```

05 04 AE 02 E3 0583 799
51 7C 0588 800
FA73 30 058A 801
68 DD 058D 802
52 FA6D CF 9E 058F 803
51 82 9A 0594 804
7E 51 7D 0597 805
50 5E D0 059A 806
51 48 AB 9E 059D 807
51 68 7D 05A1 808
FA4A 30 05B0 809
5E 08 C0 05B3 810
68 8E D0 05B6 811
05 05BC 812

```
0580 815 .SBTTL ALLOCATE AND INSERT ENTRY IN KEYPAD SYMBOL TABLE
0580 816
0580 817 :+ DCL$ALLOCKEY - ALLOCATE AND INSERT ENTRY IN KEYPAD SYMBOL TABLE
0580 818
0580 819 THIS ROUTINE IS CALLED TO ALLOCATE AND INSERT AN ENTRY IN THE KEYPAD
0580 820 SYMBOL TABLE.
0580 821
0580 822 INPUTS:
0580 823
0580 824 R6 = KEYPAD FLAGS
0580 825 R9 = ADDRESS OF BUFFER FORMATTED AS FOLLOWS
0580 826
0580 827 (R9) = DESCRIPTOR OF SYMBOL VALUE
0580 828 8(R9) = DESCRIPTOR OF SYMBOL NAME
0580 829 16(R9) = DESCRIPTOR OF SET_STATE NAME
0580 830
0580 831 R11 = ADDRESS OF PROCESS WORK AREA
0580 832 PRC_L_CURRKEY = STATE IN WHICH KEY IS TO BE ALLOCATED
0580 833
0580 834 OUTPUTS:
0580 835
0580 836 THE KEYPAD TABLE IS SEARCHED FOR THE SPECIFIED ENTRY, AND IF FOUND,
0580 837 THE OLD ENTRY IS DEALLOCATED. A SYMBOL TABLE ENTRY IS THEN ALLOCATED,
0580 838 FILLED WITH THE SYMBOL, VALUE, AND STATE INFORMATION, AND THEN
0580 839 INSERTED IN THE SYMBOL TABLE.
0580 840
0580 841 R0 LOW BIT CLEAR INDICATES ALLOCATION FAILURE WITH CLIS_SYMOVF.
0580 842 R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
0580 843
0580 844 R1,R2,R3,R4,R5 ARE DESTROYED.
0580 845
0580 846
0580 847 DCL$ALLOCKEY::
0580 848
0580 849 8E 7C 05C3 849 DISABLE (SP)+ ;DISABLE CTRL/Y'S
0580 850 05C5 850 CLRQ ;REMOVE RETURN INFO FROM STACK
0580 851
0580 852 : SEARCH FOR PREVIOUS DEFINITION OF IDENTICAL SYMBOL. IF FOUND, THEN
0580 853 DEALLOCATE IT. FIND THE SPOT IN THE LINKED LIST TO INSERT THE NEW SYMBOL
0580 854 AT.
0580 855
0580 856 51 08 A9 7D 05C5 856 MOVQ 8(R9),R1 ;SET SYMBOL NAME DESCRIPTOR
0580 857 0112 30 05C9 857 BSBW DCL$FIND_KEYPAD ;SEARCH FOR SYMBOL
0580 858 0C 50 E9 05CC 858 BLBC R0,10$ ;IF LBC SEARCH FAILURE
0580 859 7E 51 7D 05CF 859 MOVQ R1,-(SP) ;SAVE R1/R2
0580 860 30 05D2 860 BSBW DCL$DEALLOCSYM ;DEALLOCATE KEYPAD ENTRY
0580 861 51 8E 7D 05D5 861 MOVQ (SP)+,R1 ;RESTORE R1/R2
0580 862 0103 30 05D8 862 BSBW DCL$FIND_KEYPAD ;SEARCH FOR SYMBOL
0580 863
0580 864
0580 865 : CALCULATE SIZE OF NEW SYMBOL AND ALLOCATE IT.
0580 866
0580 867 51 48 AB D0 05DB 867 10$: MOVL PRC_L_CURRKEY(R11),R1 ;GET ADDR OF ASCII IF STATE
0580 868 51 51 61 9A 05DF 868 MOVZBL (R1),R1 ;GET LENGTH OF IF STATE
0580 869 51 10 A9 C0 05E2 869 ADDL 16(R9),R1 ;ADD IN SET STATE LENGTH
0580 870 51 69 C0 05E6 870 ADDL (R9),R1 ;ADD IN VALUE LENGTH
0580 871 51 04 C0 05E9 871 ADDL #4,R1 ;ADD IN SIZE OF LENGTH FIELDS
```

```
51 08 A9 DD 05EC 872      PUSHL R1      ;SAVE FOR FUTURE USE
51 0F CO 05EE 873      ADDL 8(R9),R1    ;ADD IN META-KEY LENGTH
53 DD 05F2 874      ADDL #SYM_T_SYMBOL+3,R1 ;ADD IN FIXED OVERHEAD
FA06' 30 05F3 875      PUSHL R3      ;SAVE SYMBOL TABLE PTR
53 8ED0 05F7 876      BSBW DCL$ALLDYNMEM ;ALLOCATE DYNAMIC MEMORY
47 50 E9 05FA 877      POPL R3      ;RESTORE SYMBOL TABLE PTR
      05FD 878      BLBC R0,908      ;IF LBC ALLOCATION FAILURE
      0600 879
      0600 880
      0600 881      ; INITIALIZE THE STATICALLY PLACED FIELDS AND INSERT IT IN THE LINKED LIST.
      0600 882
      08 A2 51 B0 0600 883      MOVW R1,SYM_W_SIZE(R2) ;SET SIZE OF ALLOCATED BLOCK
      08 A2 56 90 0604 884      MOVW R6,SYM_B_FLAGS(R2) ;SET KEYPAD FLAGS
      0A A2 04 90 0608 885      MOVW #SYM_K_KEYPAD,SYM_B_TYPE(R2) ;SET KEYPAD VALUE TYPE
      04 B3 62 0E 060C 886      INSQUE SYM_L_FL(R2),@SYM_L_BL(R3) ;INSERT ENTRY IN SYMBOL TABLE
      0610 887
      0610 888
      0610 889      ; INITIALIZE THE DYNAMICALLY PLACED ASCII FIELDS.
      0610 890
      53 08 A9 7D 0610 891      MOVQ 8(R9),R3 ;GET SYMBOL NAME
      0C A2 53 90 0614 892      MOVW R3,SYM_T_SYMBOL(R2) ;INSERT LENGTH OF SYMBOL
      OD A2 64 53 28 0618 893      MOVW R3,(R4),SYM_T_SYMBOL+1(R2) ;INSERT SYMBOL NAME
      83 8E F7 061D 894
      83 8E F7 061D 895      CVTLW (SP)+,(R3)+ ;INSERT LENGTH OF FOLLOWING
      52 48 AB D0 0620 896
      51 82 9A 0624 897      MOVL PRC_L_CURRKEY(R11),R2 ;GET CURRENT STATE LENGTH/ADDRESS
      83 51 90 0627 898      MOVZBL (R2)+,R1 ;GET LENGTH OF STRING VALUE
      63 62 51 28 062A 899      MOVW R1,(R3)+ ;INSERT LENGTH OF STRING VALUE
      51 69 7D 062E 900      MOVW R1,(R2),(R3) ;INSERT STRING VALUE
      83 51 B0 0631 901
      63 62 51 28 062E 902      MOVQ (R9),R1 ;GET SYMBOL VALUE
      51 69 7D 062E 903      MOVW R1,(R3)+ ;INSERT LENGTH OF STRING VALUE
      83 51 B0 0631 904      MOVW R1,(R2),(R3) ;INSERT STRING VALUE
      63 62 51 28 0634 905
      51 10 A9 7D 0638 906      MOVQ 16(R9),R1 ;GET SET STATE VALUE
      83 51 90 063C 907      MOVW R1,(R3)+ ;INSERT LENGTH OF STRING VALUE
      63 62 51 28 063F 908      MOVW R1,(R2),(R3) ;INSERT STRING VALUE
      50 01 D0 0643 909
      50 01 D0 0643 910      MOVL #1,R0 ;SET SUCCESS INDICATOR
      05 0646 911      RSB
      0647 912
      0647 913
      0647 914      ; RETURN SYMBOL TABLE OVERFLOW STATUS.
      0647 915
      8E D5 0647 916 908: TSTL (SP)+ ;RESTORE THE STACK
      05 0649 917      STATUS SYMOVF ;SET SYMBOL TABLE OVERFLOW STATUS
      05 0650 918      RSB
```



```
0651 920 .SBTTL CHECK FOR SYNONYM KEY NAMES
0651 921
0651 922 :+ DCL$SYNONYM - CHECK FOR SYNONYM KEY NAMES
0651 923
0651 924 THIS ROUTINE IS CALLED TO DETERMINE WHETHER OR NOT THE KEY NAME INPUT
0651 925 HAS A SYNONYM NAME. IF SO, IT TRANSLATES THE KEY NAME TO A COMMON KEY NAME
0651 926 FOR THAT PARTICULAR KEY, WHICH IS THEN USED IN CONSTRUCTING THE KEYPAD
0651 927 SYMBOL TABLE. IF /LOG IS SPECIFIED IN THE COMMAND LINE, A CONVERSION MESSAGE
0651 928 IS OUTPUT INDICATING WHAT THE SYNONYM KEY WAS CHANGED TO IN THE KEYPAD
0651 929 SYMBOL TABLE.
0651 930
0651 931
0651 932 INPUTS:
0651 933
0651 934 R0 = /LOG FLAG (LBS = /LOG, LBC = /NOLOG)
0651 935 R1 = LENGTH OF ENTERED KEY NAME
0651 936 R2 = ADDRESS OF ENTERED KEY NAME
0651 937
0651 938 OUTPUTS:
0651 939
0651 940 IF SYNONYM FOUND:
0651 941
0651 942 R1 = LENGTH OF TRANSLATED KEY NAME
0651 943 R2 = ADDR. OF TRANSLATED KEY NAME
0651 944
0651 945
0651 946 IF SYNONYM NOT FOUND:
0651 947
0651 948 R1 UNCHANGED
0651 949 R2 UNCHANGED
0651 950
0651 951 :-
0651 952
0651 953 DCL$SYNONYM::
0651 954 MOVQ R6,-(SP) ;SAVE WORK REGISTERS
0651 955 MOVL R0,-(SP) ;SAVE /LOG FLAG
0651 956 MOVAB SYNNAME_TAB,R6 ;GET ADDR OF SYNONYM TABLE
0651 957
0651 958 10$: MOVZBL (R6),R7 ;GET LENGTH OF THIS ENTRY
0651 959 BEQL 100$ ;EXIT IF NO MATCHING ENTRY FOUND
0651 960 CMPW R1,R7 ;DOES THE LENGTH MATCH THIS ENTRY?
0651 961 BNEQ 40$ ;NO, SKIP TO NEXT ENTRY IN TABLE
0651 962 MOVQ R1,-(SP) ;SAVE POINTERS
0651 963 CMPC5 R1,(R2),#0,R7,1(R6) ;IS THERE A MATCH ON THIS ENTRY?
0651 964 BEQL 50$ ;YES, GET NEW KEY NAME FROM TRANSLATION TABL
0651 965 MOVQ (SP)+,R1 ;RESTORE POINTERS
0651 966
0651 967 40$: MOVAB 3(R6)[R7],R6 ;MOVE TO NEXT ENTRY IN SYNONYM TABLE
0651 968 BRB 10$
0651 969
0651 970 : HAVE FOUND A MATCH IN SYNONYM TABLE. GET ACTUAL KEY NAME FROM TRANSLATION TABLE.
0651 971
0651 972 50$: MOVQ (SP)+,R1 ;RESTORE POINTERS
0651 973 MOVAB 1(R6)[R7],R6 ;GET ADDR. OF OFFSET INTO TRANS. TABLE
0651 974 MOVZWL (R6),R6 ;GET ACTUAL OFFSET
0651 975 MOVAB SYND$F_TAB[R6],R6 ;GET ADDR. OF COMMON KEY NAME STRING
0651 976 MOVQ R1,R3 ;SAVE ENTERED KEY NAME FOR CONV. MESSAGE.
```

01	A6	57	00	7E	56	7D	0651	954	MOVQ	R6,-(SP)	:SAVE WORK REGISTERS		
				7E	50	DO	0654	955	MOVL	R0,-(SP)	:SAVE /LOG FLAG		
				FA18	CF	9E	0657	956	MOVAB	SYNNAME_TAB,R6	:GET ADDR OF SYNONYM TABLE		
							065C	957					
				57	66	9A	065C	958	10\$:	MOVZBL	(R6),R7	:GET LENGTH OF THIS ENTRY	
					5A	13	065F	959		BEQL	100\$:EXIT IF NO MATCHING ENTRY FOUND	
				57	51	B1	0661	960		CMPW	R1,R7	:DOES THE LENGTH MATCH THIS ENTRY?	
					0F	12	0664	961		BNEQ	40\$:NO, SKIP TO NEXT ENTRY IN TABLE	
				7E	51	7D	0666	962		MOVQ	R1,-(SP)	:SAVE POINTERS	
				62	51	2D	0669	963		CMPC5	R1,(R2),#0,R7,1(R6)	:IS THERE A MATCH ON THIS ENTRY?	
					0A	13	0670	964		BEQL	50\$:YES, GET NEW KEY NAME FROM TRANSLATION TABL	
				51	8E	7D	0672	965		MOVQ	(SP)+,R1	:RESTORE POINTERS	
							0675	966					
				56	03	A647	9E	0675	967	40\$:	MOVAB	3(R6)[R7],R6	:MOVE TO NEXT ENTRY IN SYNONYM TABLE
					E0	11	067A	968		BRB	10\$		
							067C	969					
							067C	970					
							067C	971					
				51	8E	7D	067C	972	50\$:	MOVQ	(SP)+,R1	:RESTORE POINTERS	
				56	01	A647	9E	067F	973	MOVAB	1(R6)[R7],R6	:GET ADDR. OF OFFSET INTO TRANS. TABLE	
					56	66	3C	0684	974	MOVZWL	(R6),R6	:GET ACTUAL OFFSET	
				56	FA2B	CF46	9E	0687	975	MOVAB	SYND\$F_TAB[R6],R6	:GET ADDR. OF COMMON KEY NAME STRING	
					53	51	7D	068D	976	MOVQ	R1,R3	:SAVE ENTERED KEY NAME FOR CONV. MESSAGE.	

```
51 86 9A 0690 977      MOVZBL (R6)+,R1      ;FORM DESCRIPTOR FOR TRANS. NAME
52 66 9E 0693 978      MOVAB (R6),R2      ;
      0696 979      ;
      0696 980      ; OUTPUT CONVERSION MESSAGE IF /LOG SPECIFIED
      0696 981      ;
      22 6E E9 0696 982      BLBC (SP),100$      ;SKIP MESSAGE IF /NOLOG
SE 10 C2 0699 983      SUBL #16,SP      ;MAKE A TEMPORARY SCRATCH BUFFER
6E 51 7D 069C 984      MOVQ R1,(SP)      ;GET TRANSLATED KEY NAME
OB AE 53 7D 069F 985      MOVQ R3,8(SP)      ;GET ENTERED KEY NAME
      6E 9F 06A3 986      PUSHAB (SP)      ;SET ADDR. OF TRANS. KEY NAME
      OC AE 9F 06A5 987      PUSHAB 12(SP)      ;SET ADDR. OF ENTERED KEY NAME
51 02 D0 06AB 988      MOVL #2,R1      ;SET ARGUMENT COUNTER
50 0003DE33 8F D0 06AB 989      MOVL #CLIS KEYCNV,R0      ;SET CONVERTED KEY STATUS
      F94B' 30 06B2 990      BSBW DCL$FORMMSG      ;OUTPUT CONVERSION MESSAGE
51 6E 7D 06B5 991      MOVQ (SP),R1      ;RESTORE TRANSLATED DESCRIPTOR
SE 10 C0 06B8 992      ADDL #16,SP      ;REMOVE TEMPORARY SCRATCH BUFFER
      06B8 993
      8E D5 06B8 994 100$: TSTL (SP)+      ;REMOVE /LOG FLAG
56 8E 7D 06BD 995      MOVQ (SP)+,R6      ;RESTORE WORK REGISTERS
      05 06C0 996      RSB      ;EXIT
```

```
06C1 998 .SBTTL SEARCH FOR SYMBOL ENTRY IN KEYPAD SYMBOL TABLE
06C1 999
06C1 1000 :+ DCL$SEARCH_KEYPAD - SEARCH FOR SYMBOL ENTRY IN KEYPAD SYMBOL TABLE
06C1 1001 : THIS ROUTINE IS CALLED TO SEARCH THE KEYPAD SYMBOL TABLE FOR AN ENTRY.
06C1 1002 :
06C1 1003 :
06C1 1004 : INPUTS:
06C1 1005 :
06C1 1006 : R11 = ADDRESS OF PROCESS WORK AREA
06C1 1007 :
06C1 1008 : R1 = LENGTH OF SYMBOL.
06C1 1009 : R2 = ADDRESS OF SYMBOL.
06C1 1010 :
06C1 1011 : OUTPUTS:
06C1 1012 :
06C1 1013 : R0 = STATUS
06C1 1014 : R1/R2 = QUADWORD DESCRIBING SYMBOL VALUE:
06C1 1015 : IF R2 NONZERO, QUADWORD IS A STRING DESCRIPTOR
06C1 1016 : IF R2 ZERO, R1 IS A BINARY LONGWORD VALUE
06C1 1017 : R3 = ADDRESS OF SYMBOL ENTRY
06C1 1018 : R4 = KEYPAD FLAGS
06C1 1019 :
06C1 1020 :-
06C1 1021 :
06C1 1022 DCL$SEARCH_KEYPAD::
06C1 1023 BSBB DCL$FIND_KEYPAD ;SEARCH FOR SYMBOL ENTRY IN KEYPAD TABLE
06C1 1024 BLBC R0,10$ ;SEARCH KEYPAD SYMBOL TABLE FOR ENTRY
06C6 1025 ;IF LBC NO MATCH FOUND
06C6 1026 MOVZBL SYM_T_SYMBOL(R3),R1 ;GET LENGTH OF SYMBOL
06CA 1027 MOVAB SYM_T_SYMBOL+3(R3)[R1],R2 ;GET ADDRESS OF IF STATE LENGTH
06CF 1028 MOVZBL (R2)+,R1 ;GET LENGTH OF IF STATE
06D2 1029 MOVAB (R2)[R1],R2 ;GET ADDRESS OF VALUE LENGTH
06D6 1030 MOVZWL (R2)+,R1 ;GET LENGTH OF VALUE
06D9 1031 MOVZBL SYM_B_FLAGS(R3),R4 ;GET KEYPAD FLAGS
06DD 1032
05 06DD 1033 10$: RSB ;
```



```
06DE 1035 .SBTTL SEARCH KEYPAD SYMBOL TABLE FOR ENTRY
06DE 1036 :+
06DE 1037 : DCL$FIND_KEYPAD - SEARCH KEYPAD SYMBOL TABLE FOR ENTRY
06DE 1038 :
06DE 1039 : THIS ROUTINE IS CALLED TO SEARCH THE KEYPAD SYMBOL TABLE FOR AN ENTRY.
06DE 1040 : ONLY DEFINITIONS FOR THE PRC_L_CURRKEY STATE ARE CHECKED.
06DE 1041 :
06DE 1042 : INPUTS:
06DE 1043 :
06DE 1044 :     R1 = LENGTH OF SYMBOL NAME.
06DE 1045 :     R2 = ADDRESS OF SYMBOL NAME.
06DE 1046 :     R11 = ADDRESS OF PRC DATA STRUCTURE
06DE 1047 :     PRC_L_CURRKEY = CURRENT KEY STATE
06DE 1048 :
06DE 1049 : OUTPUTS:
06DE 1050 :
06DE 1051 :     R0 LOW BIT CLEAR INDICATES SEARCH FAILURE.
06DE 1052 :
06DE 1053 :         R1 = LENGTH OF SYMBOL NAME.
06DE 1054 :         R2 = ADDRESS OF SYMBOL NAME.
06DE 1055 :         R3 = ADDRESS OF NEXT GREATEST SYMBOL ENTRY.
06DE 1056 :         R4 ARE DESTROYED.
06DE 1057 :
06DE 1058 :     R0 LOW BIT SET INDICATES SYMBOL FOUND WITH:
06DE 1059 :
06DE 1060 :         R1 = LENGTH OF SYMBOL NAME.
06DE 1061 :         R2 = ADDRESS OF SYMBOL NAME.
06DE 1062 :         R3 = ADDRESS OF SYMBOL ENTRY.
06DE 1063 :         R4 IS DESTROYED
06DE 1064 : -
06DE 1065 :
06DE 1066 DCL$FIND_KEYPAD:: :SEARCH KEYPAD SYMBOL TABLE FOR ENTRY
06DE 1067 :
06DE 1068 :
06DE 1069 : SET ADDRESS OF SYMBOL TABLE.
06DE 1070 :
53 40 AB 9E 06DE 1071 : MOVAB PRC_Q_KEYPAD(R11),R3 :SET ADDRESS OF KEYPAD SYMBOL TABLE LISTHEAD
50 53 DO 06DE 1072 : MOVL R3,R0 :COPY ADDRESS OF SYMBOL TABLE LISTHEAD
06DE 1073 :
06DE 1074 :
06DE 1075 : SEARCH FOR THE SPECIFIED SYMBOL.
06DE 1076 :
53 63 DO 06DE 1077 10$: MOVL SYM_L_FL(R3),R3 :GET ADDRESS OF NEXT ENTRY
53 50 D1 06DE 1078 : CMPL R0,R3 :END OF TABLE?
53 38 13 06DE 1079 : BEQL 90$ :IF EQL YES
06DE 1080 :
06DE 1081 :
06DE 1082 : CHECK THAT THE SYMBOL STATE MATCHES.
06DE 1083 :
51 OF BB 06DE 1084 : PUSHR #*M<R0,R1,R2,R3> :SAVE SEARCH PARAMETERS
54 OF A3 9A 06DE 1085 : MOVZBL SYM_T_SYMBOL(R3),R1 :GET LENGTH OF SYMBOL
50 84 9E 06DE 1086 : MOVAB SYM_T_SYMBOL+3(R3)[R1],R4 :GET ADDRESS OF IF_STATE LENGTH
52 48 AB DO 06DE 1087 : MOVZBL (R4)+,R0 :GET IF_STATE LENGTH
64 50 00 51 82 9A 06DE 1088 : MOVL PRC_L_CURRKEY(R11),R2 :GET CURRENT STATE LENGTH/ADDRESS
62 51 2D 06DE 1089 : MOVZBL (R2)+,R1 :
0F BA 0702 1090 : CMPC5 R1,(R2),#0,R0,(R4) :STATES MATCH?
0708 1091 20$: POPR #*M<R0,R1,R2,R3> :RESTORE SEARCH PARAMETERS
```

```

      D9 14 070A 1092      BGTR 10$      ;IF NEQ NO
      17 19 070C 1093      BLSS 90$      ;
      070E 1094
      070E 1095      :
      070E 1096      : CHECK THAT THE SYMBOL NAME MATCHES.
      070E 1097      :
      54 0C A3 9E 070E 1098      MOVAB SYM T SYMBOL(R3),R4      ;GET ADDRESS OF SYMBOL NAME
      OF 8B 0712 1099      PUSHR #*M2R0,R1,R2,R3>      ;SAVE SEARCH PARAMETERS
      84 9A 0714 1100      MOVZBL (R4)+,R0      ;GET LENGTH OF SYMBOL NAME
      64 50 00 50 51 2D 0717 1101      CMPC5 R1,(R2),#0,R0,(R4)      ;SYMBOLS MATCH?
      62 0F BA 071D 1102      POPR #*M<R0,R1,R2,R3>      ;RESTORE SEARCH PARAMETERS
      C4 14 071F 1103      BGTR 10$      ;IF NEQ NO
      02 19 0721 1104      BLSS 90$      ;
      50 D6 0723 1105      :
      05 05 0725 1106      INCL R0      ;SET SUCCESS INDICATOR
      0726 1107      RSB      :
      0726 1108
```

```
0726 1110 .SBTTL SET KEYPAD STATE
0726 1111 :+
0726 1112 :DCL$SETKEY - SET KEYPAD STATE
0726 1113 :
0726 1114 :THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE SET KEYPAD
0726 1115 :COMMAND.
0726 1116 :
0726 1117 :INPUTS:
0726 1118 :
0726 1119 :R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.
0726 1120 :R9 = ADDRESS OF SCRATCH STACK.
0726 1121 :R10 = BASE ADDRESS OF COMMAND WORK AREA.
0726 1122 :R11 = BASE ADDRESS OF PROCESS WORK AREA.
0726 1123 :
0726 1124 :OUTPUTS:
0726 1125 :
0726 1126 :THE SPECIFIED KEYPAD STATE BECOMES THE LOCKED CURRENT STATE.
0726 1127 :-
0726 1128 :
0726 1129 DCL$SETKEY:: :SET KEYPAD SYMBOL TABLE STATE
58 01 D0 0726 1130 :MOV#1,R8 :ASSUME /LOG
56 56 7C 0729 1131 :CLR R6 :INIT STATE NAME DESCRIPTOR
072B 1132 :
072B 1133 :
072B 1134 :PROCESS THE TOKENS ON THE COMMAND LINE.
072B 1135 :
072B 1136 10$: BSBW DCL$GETDVAL :GET NEXT DESCRIPTOR VALUE
55 04 91 072E 1137 :CMPB #PTR_K_ENDLINE,R5 :END OF LINE?
30 13 0731 1138 :BEQL 50$ :BRANCH IF SO
55 03 91 0733 1139 :CMPB #PTR_K_PARAMETR,R5 :PARAMETER?
F3 13 0736 1140 :BEQL 10$ :IGNORE IF SO
F8C5' 30 0738 1141 :BSBW DCL$GETNVAL :GET QUALIFIER NUMBER
00'8F 51 91 073B 1142 :CMPB R1,#CLISK_STKY_STAT :/STATE?
08 13 073F 1143 :BEQL 20$ :YES, PROCESS IT
00'8F 51 91 0741 1144 :CMPB R1,#CLISK_STKY_LOG :/LOG?
10 13 0745 1145 :BEQL 30$ :YES, PROCESS IT
E2 11 0747 1146 :BRB 10$ :NO, GET NEXT TOKEN
0749 1147 :
0749 1148 20$: CLRQ R6 :INIT STATE NAME DESCRIPTOR
DC 53 56 7C 0749 1148 :BBS #PTR_V_NEGATE-PTR_V_FLAGS,R3,10$ :IGNORE IF NOT /STATE
00 00 E0 074B 1149 :BSBW DCL$GETDVAL :GET NEXT DESCRIPTOR VALUE
56 F8AE' 30 074F 1150 :MOVQ R1,R6 :SAVE IT AWAY
D4 11 0752 1151 :BRB 10$ :GET NEXT
0755 1152 :
0757 1153 30$: BISL #1,R8 :ASSUME /LOG
CD 58 01 C8 0757 1154 :BBC #PTR_V_NEGATE-PTR_V_FLAGS,R3,10$ :IGNORE IF NOT /NOLOG
53 00 E1 075A 1155 :BICL #1,R8 :CLEAR FLAG
58 01 CA 075E 1156 :BRB 10$ :GET NEXT
C8 11 0761 1157 :
0763 1158 :
0763 1159 :
0763 1160 :SET THE SPECIFIED STATE.
0763 1161 :
51 56 7D 0763 1162 50$: MOVQ R6,R1 :GET STATE DESCRIPTOR
24 13 0766 1163 :BEQL 90$ :EXIT IF NONE SPECIFIED
2A 10 0768 1164 :BSBB DCL$ALLOC_STATE :SET NEW STATE
1F 50 E9 076A 1165 :BLBC R0,90$ :EXIT IF ERROR
50 4C AB D0 076D 1166 :MOVL PRG_L_LASTKEY(R11),R0 :CLEAR LAST STATE
```



```
0056 30 0771 1167 BSBW DCL$DEALLOC STATE ;  
48 AB D0 0774 1168 MOVL PRC_L_CURRKEY(R11),- ;COPY KEY DEFINITION  
4C AB 0777 1169 PRC_L_LASTKEY(R11) ;  
0779 1170 ;  
0779 1171 ;  
0779 1172 ; OUTPUT LOG MESSAGE IF REQUESTED.  
0779 1173 ;  
10 58 E9 0779 1174 BLBC R8,90$ ;SKIP IF /NOLOG SPECIFIED  
48 AB DD 077C 1175 PUSHL PRC_L_CURRKEY(R11) ;SET ADDRESS OF ASCII STATE NAME  
51 01 D0 077F 1176 MOVL #1,R1 ;SET ARGUMENT COUNT  
50 0003DDD3 8F D0 0782 1177 MOVL #CLIS SETKEY,R0 ;SET STATUS  
F874 30 0789 1178 BSBW DCL$FORMMSG ;OUTPUT THE LOG MESSAGE  
078C 1179 ;  
078C 1180 90$: STATUS NORMAL ;SET NORMAL SUCCESS STATUS  
05 0793 1181 RSB ;RETURN  
0794 1182
```

			0794	1184	.SBTTL	ALLOCATE AND INIT A KEYPAD STATE SYMBOL	
			0794	1185	:	+	
			0794	1186	DCL\$ALLOC_STATE -	ALLOCATE AND INIT A KEYPAD STATE SYMBOL	
			0794	1187	:	:	
			0794	1188	THIS ROUTINE IS CALLED TO ALLOCATE AND INIT A KEYPAD STATE SYMBOL.		
			0794	1189	:	:	
			0794	1190	INPUTS:		
			0794	1191	:	:	
			0794	1192	R1/R2 = DESCRIPTOR OF NEW KEYPAD STATE		
			0794	1193	R11 = ADDRESS OF PROCESS WORK AREA		
			0794	1194	:	:	
			0794	1195	OUTPUTS:		
			0794	1196	:	:	
			0794	1197	PRC_L_CURRKEY = SET TO THE NEWLY ALLOCATED STATE SYMBOL		
			0794	1198	:	:	
			0794	1199	RO LBC INDICATES SYMBOL TABLE OVERFLOW		
			0794	1200	:	-	
			0794	1201	:	:	
			0794	1202	DCL\$ALLOC STATE::		:ALLOCATE STATE SYMBOL
			0794	1203	PDSHR	#^M<R1,R2,R3,R4,R5>	:SAVE R1-R5
			0796	1204	DISABLE		:DISABLE CTRL/Y'S
			079C	1205	MOVQ	R1,R4	:SAVE STATE DESCRIPTOR
			079F	1206	INCL	R1	:ADD ROOM FOR BYTE COUNT
			07A1	1207	BSBW	DCL\$ALLDYNMEM	:GET MEMORY TO SAVE STATE IN
			07A4	1208	BLBC	RO,90\$:BRANCH IF NO ROOM FOR SYMBOL
			07A7	1209	MOVL	R2,PRC_L_CURRKEY(R11)	:SET CURRENT KEY STATE
			07AB	1210	MOVB	R4,(R2)+	:MOVE THE STRING LENGTH
			07AE	1211	MOVC3	R4,(R5),(R2)	:MOVE THE STRING
			07B2	1212	STATUS	NORMAL	:SET NORMAL STATUS
			07B9	1213	ENABLE		:ENABLE CTRL/Y'S
			07BB	1214	POPR	#^M<R1,R2,R3,R4,R5>	:RESTORE R1-R5
			07BD	1215	RSB		:
			07BE	1216			:
			07BE	1217	90\$: ENABLE		:ENABLE CTRL/Y'S
			07C0	1218	POPR	#^M<R1,R2,R3,R4,R5>	:RESTORE R1-R5
			07C2	1219	STATUS	SYMOVF	:
			07C9	1220	RSB		:
			07CA	1221			:

```
07CA 1223 .SBTTL DEALLOCATE A KEYPAD STATE SYMBOL
07CA 1224 :+
07CA 1225 :DCL$DEALLOC_STATE - DEALLOCATE A KEYPAD STATE SYMBOL
07CA 1226 :
07CA 1227 :THIS ROUTINE IS CALLED TO DEALLOCATE A KEYPAD STATE SYMBOL.
07CA 1228 :
07CA 1229 :INPUTS:
07CA 1230 :
07CA 1231 :RO = ADDRESS OF ASCII KEYPAD STATE
07CA 1232 :R11 = ADDRESS OF PROCESS WORK AREA
07CA 1233 :
07CA 1234 :OUTPUTS:
07CA 1235 :
07CA 1236 :NONE
07CA 1237 :-
07CA 1238 :
07CA 1239 DCL$DEALLOC_STATE::
07CA 1240 :DISABLE
7E 50 7D 07D0 1241 :MOVQ R0,-(SP)
7E 52 7D 07D3 1242 :MOVQ R2,-(SP)
51 60 9A 07D6 1243 :MOVZBL (R0),R1
51 51 06 07D9 1244 :INCL R1
52 8E 30 07DB 1245 :BSBW DCL$DEADYNMEM
50 8E 7D 07DE 1246 :MOVQ (SP)+,R2
7D 07E1 1247 :MOVQ (SP)+,R0
07E4 1248 :ENABLE
05 07E6 1249 :RSB
07E7 1250 :
07E7 1251 :.END

:DEALLOCATE STATE SYMBOL
:DISABLE CTRL/Y'S
:SAVE R0-R3
:
:GET LENGTH OF TEMPORARY STATE
:INCR TO INCLUDE BYTE COUNT
:DEALLOCATE THE BLOCK
:RESTORE R0-R3
:
:ENABLE CTRL/Y'S
:
```


NULL	00000070	R	02
PRC-B-CONTINUE	000000F3		
PRC-B-DEFRADIX	000000AE		
PRC-B-EXMDEPMOD	000000AD		
PRC-B-EXMDEPWID	000000AC		
PRC-B-EXONLYL	0000012D		
PRC-B-FLAGS2	000000AF		
PRC-B-IMGFLAG	00000078		
PRC-B-OUTFLAGS	0000012C		
PRC-B-PROMPTLEN	000000F0		
PRC-C-LENGTH	00000534		
PRC-G-COMMANDS	00000133		
PRC-G-PROMPT	000000F4		
PRC-K-LENGTH	00000534		
PRC-L-CURRKEY	00000048		
PRC-L-EXMDEPADR	000000A8		
PRC-L-EXTARG	00000094		
PRC-L-EXTBLK	0000008C		
PRC-L-EXTCOD	0000009C		
PRC-L-EXTHND	00000090		
PRC-L-EXTPRM	00000098		
PRC-L-IDFLNK	000000BC		
PRC-L-IMGACTSTS	00000080		
PRC-L-INDCLOCK	0000007C		
PRC-L-INDEPTH	0000005C		
PRC-L-INDFAB	0000001C		
PRC-L-INDINPRAB	00000014		
PRC-L-INDOUTRAB	00000018		
PRC-L-INPRAB	00000008		
PRC-L-LASTKEY	0000004C		
PRC-L-LSTSTATUS	000000B0		
PRC-L-ONCTLY	000000B8		
PRC-L-ONERROR	0000006C		
PRC-L-OUTOFBAND	000000B4		
PRC-L-OUTRAB	0000000C		
PRC-L-OUTRABCTX	00000118		
PRC-L-PPFLIST	00000070		
PRC-L-RECALLPTR	0000012F		
PRC-L-RESTART	00000058		
PRC-L-SAVAP	00000000		
PRC-L-SAVFP	00000004		
PRC-L-SEVERITY	00000050		
PRC-L-SPWN	000000C0		
PRC-L-STACKLM	000000A4		
PRC-L-STACKPT	000000A0		
PRC-L-STATUS	00000054		
PRC-L-STs	00000084		
PRC-L-STV	00000088		
PRC-L-SYMBOL	00000060		
PRC-L-TMBX	00000074		
PRC-L-TRMLIST	00000010		
PRC-Q-ALLOCREG	00000020		
PRC-Q-COMMAND	000000E0		
PRC-Q-FLUSHTIME	000000D0		
PRC-Q-GLOBAL	00000028		
PRC-Q-IMAGENAME	000000D8		
PRC-Q-KEYPAD	00000040		

KEYPAD
Symbol table

F 6
- KEYPAD SYMBOL TABLE MANIPULATION ROUTI 15-SEP-1984 23:59:38 VAX/VMS Macro V04-00
4-SEP-1984 23:41:34 [DCL.SRC]KEYPAD.MAR;1

Page 32
(16)

PRC_Q_LABEL	00000030	WRK_B_MINPARN	FFFFFFFFD1
PRC_Q_LOCAL	00000038	WRK_B_PARMCNT	FFFFFFFFCE
PRC_Q_SAVEPRIV	000000E8	WRK_B_PARMSUM	FFFFFFFFCF
PRC_T_OUTDVI	0000011C	WRK_B_RECALLCNT	FFFFFFFFC5
PRC_W_ASTIOSB	000000C6	WRK_B_VALLEV	FFFFFFFFC4
PRC_W_ASTRETN	000000C8	WRK_B_VERBTYP	FFFFFFFFC2
PRC_W_ASTSTATUS	000000C4	WRK_C_INPBUSIZ	= 00000100
PRC_W_ATTMBX	0000007A	WRK_C_LENGTH	FFFFFFF486
PRC_W_FLAGS	00000068	WRK_G_BUFFER	FFFFFFF492
PRC_W_INPCHAN	00000064	WRK_G_INPBUF	FFFFFFF896
PRC_W_ONLEVEL	0000006A	WRK_G_RESULT	FFFFFFF9B6
PRC_W_OUTIFI	00000114	WRK_K_LENGTH	FFFFFFF486
PRC_W_OUTISI	00000116	WRK_L_CHARPTR	FFFFFFF48E
PRC_W_OUTMBXCHN	000000CA	WRK_L_DISALLOW	FFFFFFFE6
PRC_W_OUTMBXREF	000000CE	WRK_L_ERRORRTN	FFFFFFF9AE
PRC_W_OUTMBXSIZ	000000CC	WRK_L_EXPANDPTR	FFFFFFF486
PRC_W_PMPTCTRL	000000F1	WRK_L_IMAGE	FFFFFFFE2
PRC_W_WAITIOSB	00000066	WRK_L_MARKPTR	FFFFFFF48A
PTR_B_LEVEL	00000004	WRK_L_PAROUT	FFFFFFFFD2
PTR_B_NUMBER	00000005	WRK_L_PMPTADDR	FFFFFFF9A2
PTR_B_PARMCNT	00000006	WRK_L_PROMPTRTN	FFFFFFF9A6
PTR_B_VALUE	00000000	WRK_L_PROPTR	FFFFFFFC6
PTR_C_LENGTH	0000000C	WRK_L_QUABLK	FFFFFFFCA
PTR_K_COMMA	= 00000005	WRK_L_READRTN	FFFFFFF9AA
PTR_K_ENDLINE	= 00000004	WRK_L_RECALLPTR	FFFFFFFEA
PTR_K_LENGTH	0000000C	WRK_L_RSLEND	FFFFFFFB6
PTR_K_PARAMETR	= 00000003	WRK_L_RSLNXT	FFFFFFFBA
PTR_L_DESCR	00000000	WRK_L_SAVAP	FFFFFFF8
PTR_L_ENTITY	00000008	WRK_L_SAVFP	FFFFFFFC
PTR_V_FLAGS	= 00000014	WRK_L_SAVSP	FFFFFFF4
PTR_V_KEYWORD	= 00000015	WRK_L_SIGNALRTN	FFFFFFFFD6
PTR_V_NEGATE	= 00000014	WRK_L_SPECRTN	FFFFFFF9B2
SHOWHDR	00000000	WRK_L_TAB_VEC	FFFFFFFDE
STATE	00000069	WRK_L_VERB	FFFFFFFBE
STSSM_INHIB_MSG	= 10000000	WRK_W_FLAGS	FFFFFFF0
SYM_B_FLAGS	0000000B	WRK_W_FLAGS2	FFFFFFF2
SYM_B_NONUNIQUE	0000000B	WRK_W_IMGCHAN	FFFFFFFE
SYM_B_TYPE	0000000A	WRK_W_PMPTLEN	FFFFFFF9E
SYM_K_KEYPAD	= 00000004	_SS_	= 000000EF
SYM_L_BL	00000004		
SYM_L_FL	00000000		
SYM_M_ECHO	= 00000001		
SYM_T_SYMBOL	0000000C		
SYM_V_ECHO	= 00000000		
SYM_V_ERASE	= 00000004		
SYM_V_LOCK	= 00000003		
SYM_V_STATE	= 00000002		
SYM_V_TERMINATE	= 00000001		
SYM_W_SIZE	0000000B		
SYNDEF_TAB	000000B7		
SYNNAME_TAB	00000073		
SYSSFAO	*****		
SYSSFAOL	*****		
UNDKEY	000002FE		
VALIDATE_KEY_NAME	*****		
WRK_B_CMDOPT	FFFFFFFC3		
WRK_B_MAXPARN	FFFFFFFD0		

R 02
R 02

R 02
R 02
X 02
GX 02
R 02
X 02

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes															
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE					
\$ABSS	FFFFFFFFC (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE					
DCL\$ZCODE	000007E7 (2023.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE					

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.07	00:00:00.47
Command processing	87	00:00:00.72	00:00:03.56
Pass 1	257	00:00:09.83	00:00:28.30
Symbol table sort	0	00:00:00.87	00:00:02.66
Pass 2	216	00:00:03.21	00:00:11.48
Symbol table output	24	00:00:00.17	00:00:00.66
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	596	00:00:14.91	00:00:47.17

The working set limit was 1350 pages.
52334 bytes (103 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 538 non-local and 89 local symbols.
1251 source lines were read in Pass 1, producing 20 object records in Pass 2.
44 pages of virtual memory were used to define 28 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]SYSLDMLB.MLB;1	0
_\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	11
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	19

686 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:KEYPAD/OBJ=OBJ\$:KEYPAD MSRC\$:KEYPAD/UPDATE=(ENH\$:KEYPAD)+EXECML\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSLDMLB/LIB

0071 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INQUIRE
LIS

LEXICON
LIS

KEYPAD
LIS

LOGICAL
LIS